Weighted Key Player Problem
for Social Network Analysis

THESIS

Ryan M. McGuire, Captain, USAF

AFIT-OR-MS-ENS-11-13

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-OR-MS-ENS-11-13

Weighted Key Player Problem for Social Network Analysis

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Ryan M. McGuire, BS

Captain, USAF

March 2011

AFIT-OR-MS-ENS-11-13

Weighted Key Player Problem for Social Network Analysis

Ryan M. McGuire, BS

Captain, USAF

Approved:

//signed//	22 Mar 2011
Richard F. Deckro, DBA (Chairman)	Date
//signed//	22 Mar 2011
LTC Darryl K. Ahner, PhD (Member)	Date

AFIT-OR-MS-ENS-11-13

Abstract

Social network analysis is a tool set whose uses range from measuring the impact of marketing campaigns to disrupting clandestine terrorist organizations. Social network analysis tools are primarily focused on the structure of relationships between actors in the network. However, characteristics of the actors, such as importance or status, are generally the output of the social network analysis rather than an input. Characteristics of actors can come from a number of sources to include information gathering, subject matter experts or social network analysis. Further, the strength of relationships between actors in social networks are often assumed to be all equal. However, relationships range from strong familial like relationships to weak casual relationships. The research developed in this study uses actor characteristics, relationship strengths and location theory to identify key individuals in a social network that are strategically located to influence, intercept, strengthen or disrupt data flow between a set of actors. In this technique, actor characteristics and relationship strengths are used as inputs into the analysis and the output is a set of actors which satisfies the desired objective and the constraints of the given problem. This extends the tool set of social network analysis to targeting of actors based on actor characteristics, relationship strength and network structure.

AFIT-OR-MS-ENS-11-13

To my loving wife and son,
thank you for your love and support

Acknowledgements

I would like to thank my research advisor, Dr. Richard F. Deckro, for sharing his wisdom and knowledge with me. His guidance was instrumental in the development of this research and my graduate education. I would also like to thank my reader, LTC Darryl K. Ahner, who's thoughtful comments lead to stronger conclusions and ultimately a stronger thesis.

I would also like to thank all of my teachers throughout my life. Each of whom shaped and molded my education. I would not be the person I am today without your influences.

Finally, I would like to thank the men and women of the United States Armed Services. Your constant vigilance over our freedoms is what keeps this country free. I can only hope that this research will someday help you in keeping our great nation free.

Ryan M. McGuire

Table of Contents

List of Figures

Figure Page

List of Tables

List of Abbreviations

Weighted Key Player Problem for Social Network Analysis

I. Introduction

1.1 Background

The desire to understand individual and group behavior has existed for all of human history. This desire has manifested itself in the sciences of psychology and sociology. The uses of these sciences range from understanding human interactions, on large and small scales, to understanding the behavior of a single person. As these behavioral sciences have evolved and matured, new techniques for representing and analyzing data have arising.

One such technique is social network analysis, SNA, which is concerned with the structure of relationships between a set of actors of interest. The actors and their relationships form a social network on which analysis can be performed (Wasserman and Faust, 1994: p. 3). The measures used in SNA are primarily calculated based on the structure of the relationships between the actors being analyzed. These measures quantify how actors are connected, how many paths exist between two actors, how central an actor is, how the actors cluster and other such connection focused measures. Using these measures, an analyst is able to describe the importance and interactions of actors of interest based primarily on the structure of the relationships between those individuals and groups.

In addition to describing the current structure of a social network, SNA also allows an analyst to perform prescriptive analyses on a social network. These analyses suggest how an individual, the network or a subset of the network will respond to a given stimulus. The analyst can determine the preferred placement of the stimulus to maximize the desired effect; whether that effect is targeted at an individual, the whole network, or a subset of the network. The goal of the stimulus can be to

strengthen or weaken the network. The analyst must also take into consideration where it is practical or even possible for the stimulus to be applied in the network.

Prior to the last decade, most SNA was focused on improving the performance of the network of individuals or groups being analyzed. These networks are referred to as bright networks when they describe legal, overt groups (Raab and Milward, 2003: p. 415). Since bright networks operate in plain sight, it is often assumed that all the information about the actors and their connections are known. The term dark network refers to networks which describe illegal or covert organizations and activities (Raab and Milward, 2003: p. 415). Often, the complete structure of dark networks is not fully known. This may be due to the group practicing good operational security, OPSEC, conflicting information or limited reliable data on the organization. In the last decade, some SNA has focused on understanding and reducing the performance of dark networks (Krebs, 2002; Carley *et al.*, 2003). Dark networks have been of particular interest to the US Department of Defense (DoD) since the terrorist attacks on September 11$^{\text{th}}$.

The US DoD is interested in using SNA to help plan Information Operations (IO), predict the responses of organizations to IO, to understand the organizations that are of interest and other related analyses. One example would be using SNA to describe how a terrorist organization is structured, identifying key actors and then suggesting actions to influence them. Another objective could be identifying a group of actors that are well positioned for receiving or spreading information in the network. These types of analysis are of great interest to the DoD. However, many organizations that the DoD is interested in analyzing practice good OPSEC and would be considered a dark network. This poses problems for a military analyst using SNA tools as the complete structure of the organization may not be known.

The private sector is interested in using SNA to optimize marketing campaigns and better understand the consumer. Concepts like viral marketing rely on the connectedness of consumers. Being able to efficiently target viral ads at consumers

can greatly increase the success of a marketing campaign while at the same time keeping costs down. SNA can help identify key actors in the general public that can rapidly spread their message to a large percentage of the target market.

1.2 Problem Statement

SNA generally relies on the structure of a network of actors to perform descriptive and prescriptive analysis. Information about a specific actor is not generally considered in the existing measures and techniques used for SNA. A few techniques do incorporate non-network information about actors in their measures, but these techniques are not in wide spread use (Clark, 2005; Geffre *et al.*, 2009; Carley and Krackhardt, 1999). This means that a potential wealth of data is not being used when performing analysis on social networks.

Wasserman and Faust state that network structure is the primary focus of SNA and actor characteristics are only secondary (1994: p. 8). However, other than the few techniques referenced previously, actor characteristics are not even secondary; they are virtually non-existent in most SNA measures and techniques. This lack of actor characteristics in most SNA measures and techniques is likely due to the assumptions that underpin SNA. One of these assumptions is that the network provides opportunities and restrictions on the actions of the actors (Wasserman and Faust, 1994: p. 4). Taking this assumption to the extreme means that the characteristics of an actor do not determine their actions, only the structure of the network determines what actions an actor can and cannot take. The other assumptions that underpin SNA are listed in Section 1.4.

In addition to actor characteristics, it is common in SNA to default to the assumption that the distance or strength of a relationship between actors is unity. The rational behind this is discussed in Section 2.3.1, but the summary is that it is difficult to quantify the strength of inter-personnel relationships. However, many of the common SNA techniques allow for the use of relationship weights other than

unity. In doing so, some of the normalization techniques will no longer be valid. The purpose of normalizing the measures is for comparisons between social networks, which means that using relationship weights could potentially remove the analyst's ability to compare the results of an operation that some how changes the structure or members of the network.

This thesis develops a unique approach that addresses the problem of using node characteristics and relationship weights in the identification of a key actor or a set of key actors in a social network. This approach manifests itself as two separate problems. The first problem this thesis addresses is extending what is known as the key player problem, KPP, to include actor and relationship weights while maintaining the ability to normalize the measure (Borgatti, 2006: p. 22). The KPP consists of two subproblems, the KPP-Positive KPP-Pos and the KPP-Negative KPP-Neg. This thesis will focus on extending the KPP-Pos to include actor and relationship weights. The KKP-Pos identifies a set of key actors in a network based on their ability to reach other actors in the network using the fewest connections (Borgatti, 2006: p. 22). The second problem this thesis addresses is finding techniques that find optimal solutions to the modified KPP-Pos.

1.3 Research Objectives

The research developed in this study uses actor characteristics, relationship strengths and location theory to identify key individuals in a social network that are strategically located to influence, intercept, strengthen or disrupt date flow between a set of actors. The objective of this research can be divided into three parts. First, the KPP-Pos is extended to include actor and relationship weights while maintaining the ability to normalize the measure. Second, techniques for finding optimal solutions to the modified KPP-Pos is identified. Finally, a technique to apply the modified KPP-Pos to multi-layered social networks is developed.

The extension of the KPP-Pos needs to meet three goals. First, it needs to add the ability to weight actors as more or less important than other actors in the network. Second, it needs to add the ability to handle weighted relationships. Finally, it must do both of these things while still maintaining the ability to normalize the measure.

The techniques that this thesis investigates for finding optimal solutions to the modified KPP-Pos are p-medians from location theory and hierarchical clustering. Both techniques handle the weighting of actors and relationships. p-medians produce an answer to the KPP-Pos. Hierarchical clustering divides social network into clusters of actors that require additional processing to find the set of actors that answers the KPP-Pos. Extending previous work, both techniques allow for some actors to be designated as not eligible for being a key player.

The technique of applying the modified KPP-Pos to multi-layered social networks must be able to identify an optimal set of actors across all the layers for the modified KPP-Pos. This technique must be reproducible given the same information about the social network, the actors and the relationships. The technique may also provide a ranked set of solutions to the KPP-Pos for a decision maker to choose between.

1.4 Assumptions

Although this thesis attempts to explain all concepts needed to understand the development of the research, a basic level of knowledge is assumed. The reader should be familiar with linear algebra and basic graph theory concepts. Some knowledge of sociology would be helpful, but not required. Further, the primary focus will be on simple, undirected graphs. However, directed graphs will be mentioned when applicable. This focus on simple, undirected graphs is due to two factors. First, many social network measures only apply to undirected graphs. Second, loops do

not make sense when edges represent relationships and multiple edges will be handled using multiple layers.

This research assumes that the actors and structure of the network is known, to include weights for the edges and nodes. This is a reasonable assumption for a bright network; however, it may not be reasonable for a dark network. Additionally, the following assumptions about SNA outlined by Wasserman and Faust are assumed to hold (Wasserman and Faust, 1994: p. 4). The first assumption is that the actors in the network are dependent on each other and not independent when making decisions. Second, the interactions that the edges represent are pathways for the flow of resources. Third, the structure of the network provides opportunities or restrictions on the actions of the actors. Finally, the network represents lasting ties between the actors. These four assumptions are the basis for everything in SNA. If these assumptions are not true, then the structure of social networks would not play a role in the actions of the actors in that network. This would mean any measure based on the structure of the network would not be valid for descriptive, predictive or prescriptive analysis.

1.5 Thesis Overview

The remainder of this thesis is divided into 4 chapters. Chapter II reviews the pertinent literature in graph theory, SNA, p-medians and hierarchical clustering. Chapter III covers the methodology of solving the key player problem positive in multi-layered, weighted social networks. Chapter IV covers the analysis of data sets using the method developed in Chapter III. Finally, Chapter V provides a summary and proposes future work to extend and enhance this technique. A list of figures, tables and abbreviations can be found on pages x, xi and xii, respectively.

II. Literature Review

2.1 Introduction

This chapter reviews the relevant literature relating to graph theory, social network analysis (SNA), location theory and hierarchical clustering that supports this thesis. SNA relies on the theory and mathematics of graph theory, therefore an introduction to graph theory is provided in Section 2.2 to establish common terminology and mathematical background. SNA is then covered in Section 2.3 to include typical measures used and techniques for finding cohesive subgroups of actors. p-medians are covered in Section 2.4. Finally, hierarchical clustering is covered in Section 2.5.

2.2 Graph Theory

This section introduces the basic concepts and terminology of graph theory as it relates to social networks. Graph theory is concerned with the mathematics of sets of entities and how they are related or connected to each other. These relationships or connections are represented by a graph. A graph G consists of a vertex set $V(G)$, an edge set $E(G)$, and a relationship between an edge in $E(G)$ with two vertices in $V(G)$ (West, 2001: p. 2). A graph can be displayed pictorially by drawing the set of vertices $V(G)$ as points and the set of edges $E(G)$ as lines connecting the corresponding vertices. Figure 1 shows the graphical depiction of a sample graph. Nodes that are connected by an edge are said to be adjacent to one another.

A subgraph, H, of a graph G is a graph that only consists of vertices and edges from graph G (West, 2001: p. 6). For example, the vertices A, B and D and the edges between them from the graph in Figure 1 would be a subgraph. It is useful to discuss collections of nodes and edges in graphs that are connected together. West (2001: p. 20) defines three such collections as walks, trails and paths. A walk is defined as "a list of $v_0, e_1, v_1, \ldots, e_k, v_k$ of vertices and edges such that, for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i" (West, 2001: p. 20). A trail is a walk in

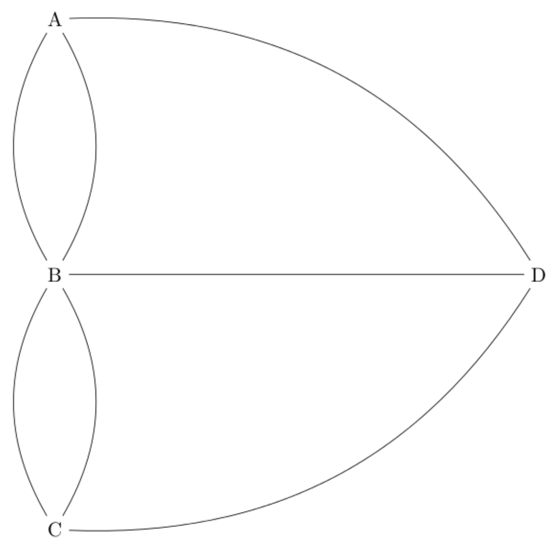

Figure 1: Graphical depiction of a sample graph

which none of the edges are repeated, however repeated vertices are allowed (West, 2001: p. 20). A path is a walk with no repeated edges or vertices. The length of a walk, trail or path is the sum of the its edges, to include repeated edges (West, 2001: p. 20). In SNA, the shortest path between actors is frequently used in calculations of centrality measures, which are covered in Section 2.3.3. Eigenvector centrality, also covered in Section 2.3.3, calculates all random walks between actors in a social network.

The edges in graphs may also carry additional information about the network, such as the distance from one vertex to another or the strength of the relationship (Wasserman and Faust, 1994: p. 140). This additional information will be referred to as an edge weight in this thesis. If an edge is weighted, the weight may be used when determining the length of a walk, trail or path. If unweighted, a length of one is used when calculating the lengths of walks, trails and paths. Vertices can also carry additional information such as capacity limit, supply or demand which will be referred to as a vertex weight in this thesis (Ahuja *et al.*, 1993: p. 203).

A graph G is said to be connected if a path exists between two vertices, i and j where $i, j \in V(G)$, otherwise it is disconnected (West, 2001: p. 21). A subgraph H is a component of a graph G if it is maximally connected (West, 2001: p. 22). If a graph is connected, it only has one component. A cut-edge or cut-vertex is an edge or vertex that when removed from a graph increases the number of components in that graph (West, 2001: p. 23). In social networks, cut-edges and cut-vertices are called bridges and cutpoints, respectively (Wasserman and Faust, 1994: p. 112, 114).

A directed graph or digraph is a graph in which the edges have a specific direction from one vertex to another (West, 2001: p. 53). A directed edge is represented by an arrow from one vertex to another vertex, where the arrow points in the direction of flow. In social networks, a directed edge represents a one-way relationship from one actor to another.

The terminology used in describing social networks is similar, but not always consistent with the terminology used in graph theory. Table 1 provides a list of terms used in graph theory and their social networks counterparts.

Table 1: Graph Theory and Social Network Terminology

Graph Theory	Social Networks
vertex	node
	actor
	point
edge	arc
	relationship
	line
component	component
cut-vertex	cutpoint
	liaison
cut-edge	bridge
adjacency matrix	sociomatrix
path	path
trail	trail
walk	walk

2.3 Social Network Analysis

This section introduces the concept of social networks and the techniques and measures used to analyze them. Many of the analysis techniques are focused on describing how important an actor is to a network and how the actors in a network are connected.

2.3.1 Social Networks. A social network is a specific type of graph that depicts relationships between actors. The actors of interest (individuals, groups, companies, and so forth) are the nodes and relationships between the actors are the arcs. To illustrate the concept of a social network, consider all the interactions with other people a single individual might have on a daily basis; whether those people are friends, coworkers, family members or acquaintances. These people, and the interactions with them, form the basis of a social network. Each person can be represented by a node on a graph and each relationship can be represented by an arc. Of course, each of the people being interacted with may also interact with other people and so on, forming a very large social network.

In 1953, Moreno first introduced the idea to represent social interactions in a network structure, called a sociogram (Moreno, 1953). Figure 2 is an illustrative sociogram for a small group of individuals that was randomly generated using the Barabási-Albert generator in Python using the NetworkX package (Barabási and Albert, 1999; Hagberg *et al.*, 2008).

A mathematical method to represent a sociogram is called an sociomatrix (Wasserman and Faust, 1994: p. 77). A sociomatrix, A, for a social network with n actors is an $n \times n$ matrix of zeros and ones. The cell $a_{i,j}$ is 1 if and only if i has a direct relationship with j and 0 otherwise. Table 2 is the sociomatrix for the social network in Figure 2. Because the social network in Figure 2 is undirected, the sociomatrix is symmetric. If a social network is directed, then the sociomatrix may be asymmetric. Asymmetric sociomatrices can result when a relationship is perceived to exist by

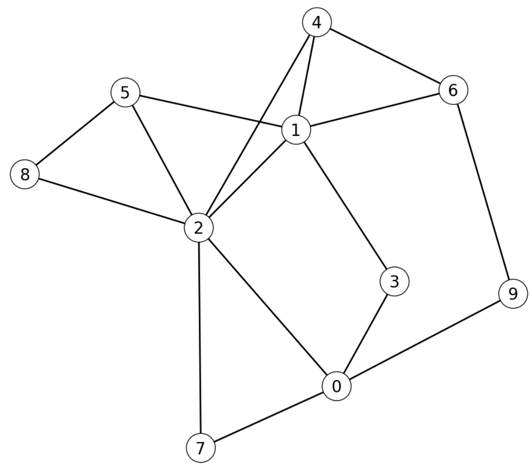

Figure 2: Sociogram for a small group of individuals

one individual, i, but not the other, j. In this example, the sociomatrix would have a 1 in the $a_{i,j}$ element, but a 0 in the $a_{j,i}$ element. Asymmetric sociomatrices can also be the result of a relationship that is only one-way. An example of this type of relationship is the at home audience members of a TV show. The audience members are receiving information from the show, but the show does not receive information from the audience members; assuming they are not a Neilson family.

Table 2: Sociomatrix for the social network in Figure 2

						j					
		0	1	2	3	4	5	6	7	8	9
	0	0	0	1	1	0	0	0	1	0	1
	1	0	0	1	1	1	1	1	0	0	0
	2	1	1	0	0	1	1	0	1	1	0
	3	1	1	0	0	0	0	0	0	0	0
i	4	0	1	1	0	0	0	1	0	0	0
	5	0	1	1	0	0	0	0	0	1	0
	6	0	1	0	0	1	0	0	0	0	1
	7	1	0	1	0	0	0	0	0	0	0
	8	0	0	1	0	0	1	0	0	0	0
	9	1	0	0	0	0	0	1	0	0	0

Social networks are not limited to relationships between individuals. Social networks can be composed of any type of relationship between any type of actor. For example, the actors might be corporations and the relationships might be payments, loans, contracts, or deliveries. Further, the actors in a social network might be of different types: individuals and companies, leaders and countries, terrorists and attack sites, and so forth. When the actors in a social network are heterogeneous, the network is called multi-mode and when composed of homogeneous actors, the network is called one-mode (Wasserman and Faust, 1994: p. 29). In this way, social networks are a very flexible representation for actors and relationships in complex systems.

A social network that only focuses on the relationships of a single actor, called the ego, is referred as an ego-centered network (Wasserman and Faust, 1994: p. 42). Typically, ego-centered networks only contain the ego and the neighbors of the ego, called alters. However, they may contain alters that are two or more relationships removed from the ego. An ego-centered network takes on the topology of a star-graph when only the neighbors of the ego are included. Figure 3 shows an ego-centered network with node 0 as the ego and nodes 1 through 6 as the alters.

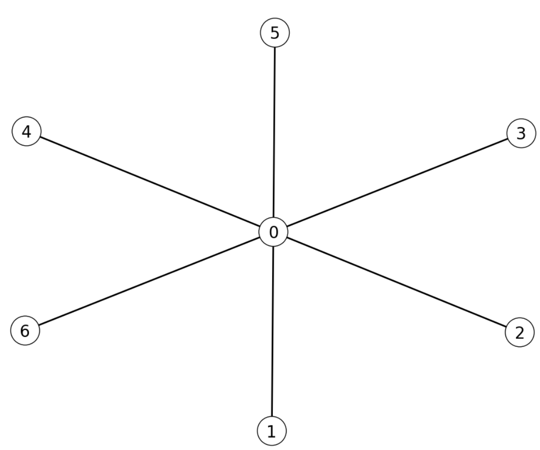

Figure 3: Ego-centered network with six alters

Although very flexible, social networks do have some limitations. It is typical to see edge or node weights in other applications of graph theory. However, social networks are dealing with interpersonal relationships that are not easily measured or quantified. Attempting to place weights on familial relationships compared to friendships illustrates this problem. To be useful the weights must be additive. For example, this means deciding the ratio of a best-friend relationship to a sibling relationship. Applying weights of nodes is equally difficult. Determining the weight of a parent compared to a best-friend is not something that can easily be measured. It may also vary depending on the context under analysis. Another problem arises once a social network is weighted. If the weights must be compared to another weighted social network, then the weighting scheme must be the same across both social networks. Without a repeatable, quantifiable process of weighting nodes and edges, most social networks have edges of length one and unweighted nodes. Techniques that address the challenges of weighting the edges in social networks have been developed, but are not in wide use (Clark, 2005; Hamill, 2006). If edges are weighted, then the sociomatrix would contain the weights rather than a 1 for adjacent actors.

2.3.2 Social Network Analysis Measures. SNA is the process of applying analysis techniques to a social network to answer specific questions about that network. Often, these questions focus around who key actors are in the social network. Other questions may be looking for groups of actors with strong ties to one another or how best to improve the communications or productivity of the group being analyzed.

The following sections cover some of the more common measures that are used in SNA. Section 2.3.3 covers the more well known centrality measures. Section 2.3.4 reviews the key player problem, specifically the key player problem positive. Section 2.3.5 covers the density measure. Section 2.3.6 covers the concept of cohesive subgroups in networks. Section 2.3.7 covers two lesser used centrality measures. Section 2.3.8 reviews techniques for dealing with multiple layers in social networks.

2.3.3 Centrality Measures. The centrality of a node in a social network is a numeric representation of how important that node is to the network. What defines "important" in the network is dependent on what question the analysis is trying to answer and what the network is representing. For this reason, there have been an array of measures developed to quantify the centrality of a node. Each measure uses different aspects of the structure of the network to calculate the node's centrality. Three of the four most popular centrality measures, degree, closeness and betweenness, were formalized by Freeman (Freeman, 1979). Throughout the review of centrality measures, Figure 2 will be used as the example social network when calculating the measures. Borgatti examined the four popular centrality measures of degree, closeness, betweenness and eigenvector to identify which measure should be used to measure a particular type of flow through a social network (Borgatti, 2005: pp. 56–63). Further, he showed that improper matching of centrality measures to flow type can result in incorrect answers (Borgatti, 2005: p. 63–69). Borgatti's results of when each centrality measure should be used based on the type of flow in the network will be reviewed at the end of this section.

The first centrality measure we will examine is degree centrality. As the name implies, it uses the degree of the actor, the number of adjacent neighbors, to determine its centrality value. Introduced in its current form by Freeman, a node's degree centrality is simply the sum of the edges incident to that node (Freeman, 1979: p. 219–221). In graph theory terms, this is the degree of the vertex. The degree centrality for node i is given in Equation 1.

$$C_D(i) = \sum_{j=1}^{n} a_{ij}, \quad i \neq j \tag{1}$$

where $a_{ij} = 1$ if i and j are adjacent and 0 otherwise

It can be seen in Equation 1 that degree centrality is dependent on the number of nodes in the network, n. This creates an upper bound of $n - 1$ on the value for

degree centrality for each node in a given network. To normalize degree centrality, the maximum possible degree of the network, $n-1$, is used as shown in Equation 2. This allows for comparison of degree centrality between nodes from different social networks of varying sizes.

$$C'_D(i) = \frac{\sum_{j=1}^{n} a_{ij}}{n-1}, \quad i \neq j \tag{2}$$

where $a_{ij} = 1$ if i and j are adjacent and 0 otherwise

If the social network is directed, the degree centrality can be divided into two parts, in-degree and out-degree centrality. In-degree centrality, $C_{D^-}(i)$, counts the number of relationships coming into actor i. Out-degree centrality, $C_{D^+}(i)$, counts the number of relationships going out of actor i. Equations 3 and 4 are the equations for in-degree and out-degree for actor i, respectively.

$$C_{D^-}(i) = \sum_{j=1}^{n} a_{ij}^-, \quad i \neq j \tag{3}$$

where $a_{ij}^- = 1$ if i is adjacent to j and 0 otherwise

$$C_{D^+}(i) = \sum_{j=1}^{n} a_{ij}^+, \quad i \neq j \tag{4}$$

where $a_{ij}^+ = 1$ if j is adjacent to i and 0 otherwise

Equations 5 and 6 are the equations for normalized in-degree and normalized out-degree for actor i.

$$C'_{D^-}(i) = \frac{\sum_{j=1}^{n} a_{ij}^-}{n-1}, \quad i \neq j \tag{5}$$

where $a_{ij}^- = 1$ if i is adjacent to j and 0 otherwise

$$C'_{D^+}(i) = \frac{\sum_{j=1}^{n} a_{ij}^+}{n-1}, \quad i \neq j \tag{6}$$

where $a_{ij}^+ = 1$ if j is adjacent to i and 0 otherwise

By calculating the normalized degree centrality for all the nodes in the social network in Figure 2, the values shown in Table 3 are obtained. The values have been rounded to five digits. Based on this measure, nodes 0, 1 and 2 are considered to

Table 3: Normalized degree centrality for social network in Figure 2

Node	Degree Centrality
0	0.44444
1	0.55556
2	0.66667
3	0.22222
4	0.33333
5	0.33333
6	0.33333
7	0.22222
8	0.22222
9	0.22222

be most central to this network. Looking at the sociogram in Figure 2, without any other information about the type of flow in the network, this seems to be a reasonable assessment for this network. These three nodes have the most connections to other nodes. Nodes 3, 7, 8 and 9 have the fewest connections and therefore have the lowest degree centrality.

Eigenvector centrality is defined as the principal eigenvector of the sociomatrix of the social network (Bonacich, 1972). Eigenvector centrality accounts for how connected the actors are that are adjacent to a given actor. This means that although a given actor, i, may only be connected to one other actor, j, if actor j is well connected, then actor i will have a high eigenvector centrality. Eigenvector centrality attempts to capture the significance of being connected to actors that are themselves highly connected. An eigenvector, v, is defined by

$$Av = \lambda v$$

where A is the sociomatrix, v is the eigenvector, and λ is a constant. It is not always possible to calculate the eigenvector centrality for a social network because eigenvalues do not exist for the sociomatrix, A.

Calculating the normalized eigenvector centrality for all the nodes in the social network in Figure 2, the values shown in Table 4 are obtained. The values have been rounded to five digits. Based on this measure, nodes 1, 2, 4 and 5 are considered

Table 4: Normalized eigenvector centrality for social network in Figure 2

Node	Eigenvector Centrality
0	0.29912
1	0.44758
2	0.50994
3	0.20500
4	0.33323
5	0.32588
6	0.25623
7	0.22212
8	0.22947
9	0.15247

to be most central to this network. Looking at the sociogram in Figure 2, without any additional information about the flow in the social network, this seems to be a reasonable assessment for this network. Nodes 1 and 2 can influence the most number of nodes and nodes 4 and 5 are able to influence both nodes 1 and 2. All the other nodes, with the exception of node 9, can either influence node 1 or 2, but not both. Node 9 cannot influence either node 1 or 2, resulting in the lowest eigenvector centrality.

Betweenness centrality is a measure of how often an actor is on the shortest path between two other actors (Freeman, 1979: p. 221–224). This measure attempts to quantify how much an actor can control the flow of information or goods between other actors in the network. This assumes that information or goods flowing through the social network are following the shortest paths. The measure requires the calcu-

lation of all shortest paths between all actors in the social network. If more than one shortest path exists for a set of actors, each one must be found. This can become quite computationally intensive as the size of the network increases. We define g_{ij} as the number of shortest paths, also called geodesics, between actors i and j and g_{ikj} is defined as the number of shortest paths between i and j that include k. The betweenness centrality of actor k is then given by Equation 7. Equation 8 gives the normalized betweenness centrality for actor i. The term used to normalize the measure in Equation 8 comes from a proof by Freeman (1977: p. 38) in which he proves that the maximum betweenness centrality that any graph may take is $\frac{n^2-3n+2}{2}$.

$$C_B(k) = \sum_i \sum_j \frac{g_{ikj}}{g_{ij}}, \quad i \neq k \neq j \tag{7}$$

$$C'_B(k) = \frac{2C_B(k)}{n^2 - 3n + 2} \tag{8}$$

Calculating the normalized betweenness centrality for all the nodes in the social network in Figure 2, the values shown in Table 5 are obtained. The values have been rounded to five digits. Based on this measure, nodes 0, 1 and 2 are considered to be most central to this network. Looking at the sociogram in Figure 2, this again seems to be a reasonable assessment for this network without any other information. These

Table 5: Normalized betweenness centrality for social network in Figure 2

Node	Betweenness Centrality
0	0.18519
1	0.21296
2	0.35111
3	0.01389
4	0.03241
5	0.03241
6	0.06944
7	0.00000
8	0.00000
9	0.03704

three nodes lie on the most shortest paths between pairs of nodes in this network. Nodes 7 and 8 do not lie on any of the shortest paths between other nodes, so they have a betweenness centrality of 0.

While betweenness centrality is a measure of being on the most number of shortest paths between sets of actors, closeness centrality is a measure of being as close to all other actors as possible (Freeman, 1979: p. 224–226). In this position, an actor needs to rely on the fewest number of people to send or receive a resource flowing in the network. The closeness of node i is defined by Equation 9.

$$C_C(i) = \sum_{j=1}^{n} d_{ij}, \quad i \neq j \tag{9}$$

where d_{ij} is the shortest distance between nodes i and j

Similarly to betweenness centrality, closeness centrality requires the calculation of the shortest path between actors. However, only one shortest path to all other actors must be found for each actor in the network. Closeness centrality, like betweenness and degree centrality, can be normalized. The normalized closeness centrality for actor i is given in Equation 10.

$$C'_C(i) = (n-1)C_C(i) \tag{10}$$

It is important to note that closeness centrality can only be calculated for a connected social network. If the network is not connected, then $d_{ij} = \infty$ when j is not reachable from i. In the case of a disconnected social network, closeness centrality can be calculated for each of the components of the network. This extends to directed graphs, which must be strongly connected to calculate closeness centrality.

Calculating the normalized closeness centrality for the nodes in the social network in Figure 2, the value shown in Table 6 are obtained. The values have been rounded to five digits. Based on this measure, nodes 0, 1 and 2 are considered to be

Table 6: Normalized closeness centrality for social network in Figure 2

Node	Closeness Centrality
0	0.64286
1	0.69231
2	0.75000
3	0.52941
4	0.60000
5	0.56250
6	0.52941
7	0.52941
8	0.47368
9	0.50000

most central to this network. Looking at the sociogram in Figure 2, this seems to be a reasonable assessment for this network without more information about the flow in the network. These three nodes can reach all other nodes in the fewest number of steps through the network. Nodes 8 and 9 are the furthest from all other nodes and therefore have the lowest closeness centrality.

A summary of the ranking for each actor in Figure 2 for each of the centrality measures discussed above are given in Table 7. Three of the four measures, degree, eigenvector and closeness, had the same rankings for the three most central actors in the network, nodes 0, 1 and 2. Eigenvector had the same rankings for the two most important actors as the other three measures. A Friedman test was preformed on the rankings in Table 7 with the null hypothesis being that the ranks of the centrality measures are the same and the alternative hypothesis being that at least one is different. The test statistic was 0.12000 and the p-value was 0.98933, both rounded to 5 digits. This indicates that the ranks of the four measures are not statistically different. This similarity in centrality measure rankings will not always hold and is dependent on the structure of the network being analyzed. Generally, only one centrality measure should be selected for an analysis. This selection should be based on the type of flow being assumed to exist in the social network. For this

reason, it is important to understand what types of flow each centrality measure is assuming. The following covers the flow assumptions for each centrality measure.

Table 7: Ranking of actors in Figure 2 by centrality measure

	Centrality Measure			
Actor	Degree	Eigenvector	Betweenness	Closeness
0	3	5	3	3
1	2	2	2	2
2	1	1	1	1
3	5	9	7	6
4	4	3	6	4
5	4	4	6	5
6	4	6	4	6
7	5	8	8	6
8	5	7	8	8
9	5	10	5	7

Borgatti (2005) examined the assumptions of each of the centrality measures covered in Section 2.3.3 and determined the type of flow through a network that it models. There are two attributes that can determine the type of flow that is occurring in the network (Borgatti, 2005: p. 58–59). The first attribute is the type of replication or transmission that the network allows the flow to use when spreading from one actor to the next. The transmission types that Borgatti identifies are parallel duplication, serial duplication and transfer. In parallel duplication, the flow is allowed to spread from one actor to all other connected actors, such as a speech. Serial duplication is only allowed to pass from one actor to one other connected actor, such as a rumor. Finally, transfer deals with the flow of indivisible items like books or packages. The second attribute is the type of trajectory that the flow follows in the network. Borgatti suggests four trajectories that flow can follow: geodesics, paths, trails and walks, each of which corresponds to a graph theory term.

Borgatti likens degree centrality to measuring flow in a network at time $t + 1$ (Borgatti, 2005: p. 62). This is because degree centrality only deals with the neigh-

bors of the node, those that would be affected next by the flow. Degree centrality works for parallel duplication flows because it is concerned with all adjacent actors (Borgatti, 2005: p. 62). Degree centrality then is a good measure for immediate influence of an actor over members of the network. Using degree centrality to measure the long term affects of a flow or to determine who controls the flow of information would result in incorrect results (Borgatti, 2005: p. 63–69).

Eigenvector centrality does not make any restrictions on the trajectory that the flow in the network must make, so the flow follows walks through the network (Borgatti, 2005: p. 62). Further, it allows for parallel duplication of the flow in the network. Based on this, Borgatti concludes that eigenvector centrality is well suited to measure a nodes ability to influence the rest of the network (Borgatti, 2005: p. 62).

Closeness centrality is a good measure for processes in which the flow follows the shortest path or the flow allows parallel duplication (Borgatti, 2005: p. 59–60). Examples of these types of flows are package delivery systems, attitude influencing and broadcast messaging like email (Borgatti, 2005: p. 59).

Betweenness centrality is a good measure for systems that behave like a package routing system (Borgatti, 2005: p. 61). The system knows the shortest route to take and has a starting and ending point. Further, the flow is indivisible and must take only one shortest path if multiple shortest paths exist. He concludes that this measure is inappropriate for measuring gossip, influence, infection, or information flow in a network (Borgatti, 2005: p. 61).

The flows that Borgatti described and which centrality measure is appropriate for each is summarized in Table 8. It should be noted that there are many types of flow that do not have a measure that can accurately model their behavior. Some of these missing measures are for flows that might be of interest in SNA, such as gossip, emotional support and viral infection (Borgatti, 2005: p. 59, 63).

Table 8: Flow types and centrality measures (Borgatti, 2005: p. 63)

	Parallel duplication	Serial duplication	Transfer
Geodesics	—	Closeness	Closeness Betweenness
Paths	Closeness Degree	—	—
Trails	Closeness Degree	—	—
Walks	Closeness Degree Eigenvector	—	—

2.3.4 Key Player Problem. The centrality measures covered previously attempt to quantify how important a single actor is to the network. However, if a set of more than one important actor is needed, these measures may not provide the best answer. Borgatti (2006: p. 22) defines this problem as the key player problem, KPP. He divides the problem into two separate subproblems, the KPP-Positive, KPP-Pos, and the KPP-Negative, KPP-Neg. As mentioned previously, the KPP-Pos identifies sets of actors that can reach all other actors in the network using the fewest number of connections (Borgatti, 2006: p. 22). The KKP-Neg identifies sets of actors that if removed from the network would maximally separate the network (Borgatti, 2006: p. 22). For these two subproblems, he asserts that the standard centrality measures are not effective at determining an optimal solution, a set of key players.

Equation 11 is how Borgatti defines the measure for KPP-Pos, ${}^{D}R$, where d_{Kj} is the minimum distance from all key players to actor j (Borgatti, 2006: p. 29).

$$ {}^{D}R = \frac{\sum_j \frac{1}{d_{Kj}}}{n} \tag{11} $$

This formula assumes that a key player is distance 1 from itself, $d_{ii} = 1$ (Borgatti, 2006: p. 29). One reason for this is so that dividing by the number of actors

in the network, n would normalize the measure. This violates the graph theory concept that a vertex is distance 0 from itself. Another solution would have been to divide by $n-k$, where k is the size of the key player set and to leave $d_{ii}=0$. This solution is shown in 12 and needs the requirement that $n>k$, which is a reasonable assumption.

$$^{D}R = \frac{\sum_j \frac{1}{d_{Kj}}}{n-k} \tag{12}$$

Hamill discusses using the KPP-Pos to identify actors in a social network for the purpose of developing target sets (Hamill, 2006: p. 176–179). He notes that the normalization is not on the range $[0,1]$, but instead $[k/n,1]$ (Hamill, 2006: p. 162).

2.3.5 Density. The density of a social network is a measure of the number of relationships or edges that exist in the social network compared to how many could possibly exist (Wasserman and Faust, 1994: p. 101–103). The total number of edges that could exist is a function of the number of actors, n, in the social network. The total number of possible edges in a social network is given by $\binom{n}{2}$ or $n(n-1)/2$. The density of a social network, Δ, is given by Equation 13.

$$\Delta = \frac{2E}{n(n-1)} \tag{13}$$

where E is the number of edges in the social network

The density of a social network can be thought of as how close the network is, in percentage, to a complete graph. A density of 1 indicates that the social network is a complete graph, while a density of 0 means there are no relationships in the social network. Another view of density is that it is the average proportion of relationships that each actor participates in (Wasserman and Faust, 1994: p. 102).

2.3.6 Cohesive Subgroups. While centrality and KPP are focused on finding key actors and sets of key of actors in a social network, the techniques discussed in this section are focused on finding groups of actors that are strongly connected. Formally, "cohesive subgroups are subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties" (Wasserman and Faust, 1994: p. 249). Identifying cohesive subgroups within a social network provides the analyst with groups of actors that may be very similar to each other in beliefs or actions (Collins, 1988: pp. 416–417). The colloquial term for a group of people that have strong ties is a clique. Commonly, this term is used when describing the subgroups that form in schools. SNA uses this term, along with clans, clubs, k-plexes and k-cores to describe different types of cohesive subgroups in social networks (Wasserman and Faust, 1994: p. 254, 260–261). Each of these terms will be discussed in this section.

This section focuses on cohesive subgroups in one-mode networks. There are techniques for finding and analyzing cohesive subgroups in affiliation networks, a special type of two-mode network, but they will not be covered in this study. Chapter 8 of Wasserman and Faust (1994) provides a thorough review of cohesive subgroups in affiliation networks.

We begin our overview of cohesive subgroups with the simplest structure, the clique. As mentioned before, this term is used both in common speech and in SNA to describe a group of actors that all know each other. The more formal definition of a clique is a complete graph of three or more nodes (Wasserman and Faust, 1994: p. 254). Figure 4 shows a clique of six actors. Each actor in Figure 4 has a relationship with every other actor, thus forming a clique. Cliques may exist as subgraphs in social networks.

If even one edge is missing from the group of actors in Figure 4, the group of actors would not be considered a clique. In the example of a single edge missing from Figure 4, the two actors that are not directly connected will be a distance of 2 from

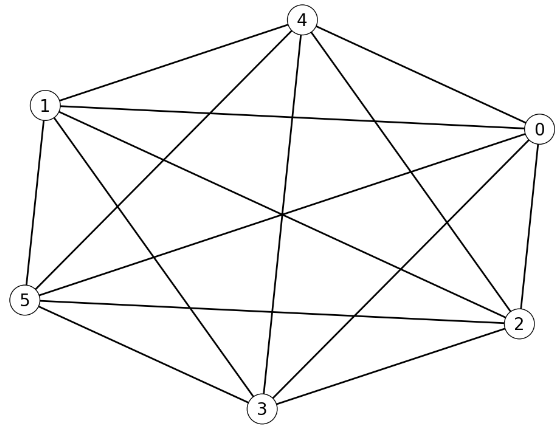

Figure 4: A clique of six actors

each other. This situation requires a less strict subgroup definition. n-cliques use the geodesic distance between actors to define a subgroup (Wasserman and Faust, 1994: p. 258). An n-clique is a subgraph, G_s, that contains nodes from the original graph, G, such that $d(i,j) \leq n,$ for all $i,j \in G$ and there are no other nodes in G that meet this requirement (Wasserman and Faust, 1994: p. 258). As mentioned in Section 2.3.1, the weights on the edges are assumed to be 1. The social network in Figure 5 has two sets of 2-cliques: {1,2,3,4,5} and {2,3,4,5,6}. The actors in these sets are, at most, a distance of two from each other in the original network. However, within the subgraph that node set {1,2,3,4,5} forms, nodes 4 and 5 are distance 3 from each other. The definition of n-cliques allows for actors to use nodes and edges outside the n-clique to determine the shortest distance between them. This can result in n-cliques with diameters larger than n, which is not always desirable.

The n-clan and n-club are types of cohesive subgroups that try to deal with the n-clique problem of diameters larger than n (Wasserman and Faust, 1994: p. 260-262). Both are based on the idea of reachability in the subgraph. n-clans are proper subsets of n-cliques that only contain subgraphs, G_s, such that $d(i,j) \leq n$, for all $i,j \in G_s$. This means that all n-clans are n-cliques. In Figure 5, the

only 2-clan is {2,3,4,5,6}. n-clubs are defined as "maximal subgraphs of diameter n" (Wasserman and Faust, 1994: p. 261). This means that not all n-clubs are n-cliques, because n-clubs exclude nodes that n-cliques must include by definition. In Figure 5, there are three 2-clubs: {1,2,3,4}, {1,2,3,5} and {2,3,4,5,6}.

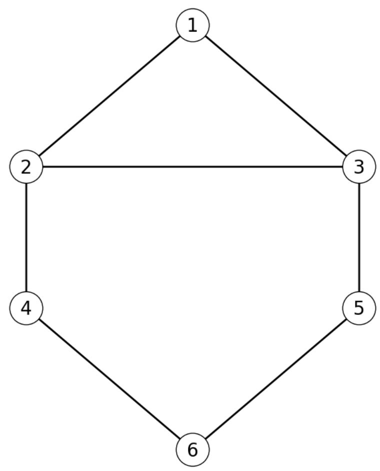

Figure 5: Graph illustrating n-cliques, n-clans and n-clubs (from Wasserman and Faust, 1994: p. 259)

k-plexes and k-cores are cohesive subgroups that are based on a minimum adjacency to other members of the subgroup (Wasserman and Faust, 1994: p. 263). k-plexes are "maximal subgraphs containing g_s nodes in which each node is adjacent to no fewer than $g_s - k$ nodes in the subgraph" (Wasserman and Faust, 1994: p. 265). This allows the formation of a subgraph of actors that each may have up to k edges missing between all other actors. On the other hand, k-cores are defined by each node having a minimum number of edges, k, to other members of the k-core. This means that a node must have at least k adjacent nodes in the subgraph to be part of the k-core.

2.3.7 Katz's Influence and Information Centrality. This section will cover two additional SNA measures: Katz's influence and information centrality. They are

not as widely used in SNA as the measures covered earlier, however they can provide a different insight into the actors and structure of the network.

Katz was not satisfied by the common popularity contest type status measures that were available to him (Katz, 1953: p. 39). He devised a measure of status that was not concerned with how many choose who in the network, but who chooses who in the network and the status of the chooser. Of course, the chooser's status is based on who chose them and so on. This creates an infinite loop of choosers, with no actor having any status to begin the calculation. This can be resolved using linear algebra to solve Equation 14, where a is an attenuation factor, C is the adjacency matrix, t is column vector of the column sums of $(I - aC)^{-1} - I$ and s is a column vector of the column sums of C.

$$\left(\frac{1}{a} - C'\right) t = s \tag{14}$$

Katz notes that there is a restriction on the value of $\frac{1}{a}$ that can be used in his measure. He recommends using a value between the largest root of C and two times that value. Katz's influence measure is similar in nature to eigenvector centrality (Borgatti, 2005: 61).

Information centrality is similar to betweenness centrality which was discussed in Section 2.3.3; however, information centrality considers all paths between actors and weights each path based on its length (Wasserman and Faust, 1994: p. 193). The weights of the paths are assigned using the inverse of the path length. The following steps are outline by Wasserman and Faust (1994: p. 195-196) to calculate information centrality for an actor i. First, construct a matrix, A, which has diagonal elements

$$a_{ii} = 1 + \text{ sum of values for all edges incident to } i$$

remembering that values for the edges are 1 for this thesis. The off-diagonal elements of A are given by

$$a_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are not adjacent} \\ 1 - x_{ij} & \text{if nodes } i \text{ and } j \text{ are adjacent} \end{cases}$$

where x_{ij} is the weight of the edge between actors i and j. For the example in this study, the weight of the edges between actors is 1 because no information about relationship strength was given. The inverse of matrix A, which will be called C, is now calculated. Next, sums of the elements of C are calculated as follows: $T = \sum_{i=1}^{n} c_{ij}$ and $R = \sum_{j=1}^{n} c_{ij}$. Finally, the information centrality for an actor i is calculated using Equation 15.

$$C_I(i) = \frac{1}{c_{ii} + (T - 2R)/n} \tag{15}$$

Table 9 is the information centrality for the social network in Figure 2. Nodes 0, 1 and 2 are ranked highest in information centrality, while nodes 3, 7, 8 and 9 are the lowest. When compared with the ranks of the centrality measure in Table 7, information centrality ranks the nodes in this example in similar order to degree, betweenness and closeness centrality.

2.3.8 Multi-Layered Networks. As mentioned in the example in Section 2.3, ones interactions with other people include family, friends, co-workers, acquaintances and possibly other contexts for a relationship. It is also possible that between two actors, more than one relationship context might exist. For example, a co-worker might also be a friend or a boss might also be an in-law. In SNA, when more than one context for relationships are considered, the analysis is called multiplex (Monge and Contractor, 2003: p. 35).

Table 9: Information centrality for social network in Figure 2

Node	Information Centrality
0	1.41490
1	1.58480
2	1.68432
3	1.05110
4	1.27396
5	1.21958
6	1.22258
7	1.03491
8	0.97700
9	1.00343

When dealing with multiplex data, the context of a relationship can affect how much influence that relationship will have over an actor. It might seem obvious, but a friend probably has more influence over an individual than an acquaintance. The context of relationships can also be affected by the actor's culture or religion. Further, the influence of a context for one actor might be vastly different for another actor.

Previously in this thesis when adding a relationship to the sociomatrix, the context of that relationship, such as friendship, familial, or others, was not considered. Now, however, we will maintain separate sociomatrices for each relationship context. Wasserman and Faust (1994) suggested the notation x_{ijr} for the existence or strength of a relationship between actors i and j in context r. The sociomatrix, A will have a 1 in cell a_{ijr} if and only if actors i and j have a relationship in context r and 0 otherwise. If relationship weights are being used, then the weight w_{ijr} is used for a_{ijr}. In this form, the sociomatrix is sometimes referred to as a super-sociomatrix (Wasserman and Faust, 1994: p. 81).

The three dimensional structure of a super-sociomatrix lends itself to being displayed as a multi-layered network. For a given value R of the relationship context r, the resulting matrix, A_{ijR}, is a standard sociomatrix which can be visualized.

Doing this for all values of r results in a visualization of each layer in our multi-layered network.

Some of the techniques introduced by various authors for dealing with multiplex data is now introduced. The focus will be on the concept of the technique, leaving the details to the references. This is for two reasons. First, in general, most SNA has been limited to analysis of a single context (Bonacich *et al.*, 2004: p. 189). Secondly, some of the following techniques can be very involved and require extensive examples to illustrate, which are best left to the references.

Wasserman and Faust (1994) suggest not combining the layers of the super-sociomatrix into one sociomatrix, as suggested by Knoke and Burt (1983). Instead, they suggest performing independent analysis on the individual layers of the super-sociomatrix (Wasserman and Faust, 1994: p. 219). Hamill (2006) postulates that this might be due to the loss of information that occurs when a super-sociomatrix is collapsed into a single level. Losing the information about what context the relationships came from and aggregating them into a single numeric value can introduce issues when comparing relationship strengths in the network. This technique requires the analyst to perform the final stage of the analysis without the aid of a repeatable, mathematical process or to only draw conclusions from the independent layers.

Carley suggests combining the sociomatrix with additional network data, like tasks and resources, to produce a metamatrix (Carley and Krackhardt, 1999: p. 2). Unlike a super-sociomatrix, the metamatrix contains the networks, capabilities, assignments, substitutes, needs and precedence submatrices that are defined by the personnel, tasks and resources (Carley and Krackhardt, 1999: p. 3). Carley and Krackhardt (1999) uses the metamatrix for forming a typology for classifying network measures based on the submatrix that should be used for that measure. It is clear that this approach is useful for analysis of organizations that perform tasks with resources; however, this technique may not be well suited for analyzing more informal networks like friends and family.

Multidimensional centrality, MDC, is a technique that can measure how important a context is in a super-sociomatrix (Bonacich *et al.*, 2004: p. 202). MDC is similar to eigenvector centrality in that it calculates the eigenvector of a matrix of hyper edges. Clark (2005) uses MDC to calculate the weights, w_i, for each context, i, in a linear model of the form

$$W = w_1 I_1 + w_2 I_2 + \cdots + w_n I_n$$

$$\sum_{i=1}^{n} w_i = 1$$

where I_i is a matrix of pair-wise measures in context i, to combine the layers of a super-sociomatrix into a single sociomatrix. Using this technique, Clark is able to create a single sociomatrix that includes the influence from all layers of the super-sociomatrix. Problems with this exact implementation have been raised due to the use of information centrality as the pair-wise measure in the I matrix (Hamill, 2006: p. 200–202). However, this method, with some modification, allows a super-sociomatrix to be reduced to a sociomatrix with asymmetric arc weights between actors.

Hamill (2006) develops a technique that, while similar to Clark's technique, uses subject matter experts to provide initial weights for the contexts of the super-sociomatrix and numerical estimation techniques to refine those weights. This technique overcomes the problems that Hamill found in Clark's method, while leveraging subject matter experts to weight the importance of each context based on the group under analysis.

2.4 p-median

In location theory, the p-median is defined as a set of p vertices that minimize the sum of distances from all other vertices to their closest vertex in the set of p vertices (Minieka, 1977: p. 648). The measure used for distance is not restricted

to a physical distance, but could represent any quantifiable difference between the vertices in the graph. The integer program formula of the p-median problem is given in Equations 16–20 (Marianov and Serra, 2009: p. 4).

$$\min \sum_{i=1}^{N} \sum_{j=1}^{M} h_i d_{ij} x_{ij} \tag{16}$$

$$\text{subject to: } \sum_{j=1}^{M} x_{ij} = 1 \quad i = 1, \ldots, N \tag{17}$$

$$\sum_{j=1}^{M} y_j = p \tag{18}$$

$$x_{ij} \leq y_j \quad i = 1, \ldots, N, \; j = 1, \ldots, M \tag{19}$$

$$x_{ij}, y_j \in 0, 1 \quad i = 1, \ldots, N, \; j = 1, \ldots, M \tag{20}$$

In Equations 16–20, the parameter h_i is the weight of vertex i and d_{ij} is the distance from vertex i to vertex j. The variable $x_{ij} = 1$ if vertex i is assigned to vertex j and 0 otherwise and $y_j = 1$ if vertex j is a median of the graph and 0 otherwise. Equation 16 is the objective function which minimizes the summation of the weighted distance from vertex i to its assigned vertex j. Equation 17 limits a vertex i from being assigned to more than 1 vertex j. Equation 18 limits the number of vertices selected to be medians to the value p. Equation 19 prevents a vertex i from being assigned to a vertex j that in the set of medians. Equation 20 restricts the variables to be either 0 or 1.

Hamill (2006: p. 175–179) discusses using the p-median to solve for optimal sets of key actors in the KPP-Pos. He formulates an unweighted and weighted version. His formulation of the weighted p-median for solving the KPP-Pos is discussed in Section 3.4.

2.5 *Hierarchical Clustering*

Hierarchical clustering is a technique for grouping a set n items together in a manor that attempts to minimize a given objective function at each step (Ward Jr, 1963: p. 236). At each step the number of items is reduced by 1 until all items have been grouped together. The objective function is a function that is defined by the analyst such as distance, information loss, or any other function that can be used to determine how good a grouping is compared to all other grouping.

Starting with a group of n items, the objective function is calculated for all possible combinations of joining two items into a single group or cluster. The grouping with the lowest objective function value is selected and clustered. This results in $n-1$ items remaining in the group, the $n-2$ single items and the 1 cluster. Again, each combination of groupings for the $n-1$ items is considered and the grouping with the lowest objective function is selected and clustered. This process continues until only one cluster remains which contains all n items. There are a number of techniques for handling the objective function for an existing cluster and one of the remain items or between existing clusters. These include average, complete, closest point and others. The technique used in this thesis is average. The function for the complete method is shown in Equation 21 and the function for the average method is shown in Equation 22, where u and v are the clusters or items under consideration, i and j are the individual items in each cluster, dist is the objective function measure being used between items and clusters, and $|u|$ and $|v|$ are the cardinality of u and v.

$$d(u, v) = \max(\mathrm{dist}(u[i], v[j])) \quad \forall\, i \in u,\ j \in v \tag{21}$$

$$d(u, v) = \sum_{ij} \frac{\mathrm{dist}(u[i], v[j])}{|u| * |v|} \tag{22}$$

Hierarchical clustering can be used to form groups of similar actors in a social network Wasserman and Faust (1994: p. 381). The weights on the relationships between actors can be used in the objective function. This thesis will use the shortest path between actors and clusters as the measure for the objective function.

Building on the information reviewed in this chapter, Chapter III extends the KPP-Pos measure to include node and edge weights. Following that, the use of the p-median and hierarchical clustering to find an optimal solution to the extended KPP-Pos is discussed. Following that, the application of the extended KKP-Pos to multi-layered social networks is covered.

III. Methodology

3.1 Introduction

This chapter discusses the methodology of solving the key player problem positive, KPP-Pos, in multi-layered, relationship and actor weighted social networks. The latter is referred to as the weighted KPP-Pos, WKPP-Pos. To do this, the KPP-Pos is expanded to include relationship and actor weights while still allowing for normalization. The use of the p-median and hierarchical clustering are discussed to identify the key player sets that maximize the WKPP-Pos measure. Figure 6 illustrates the methodology developed in this study to solve the WKPP-Pos. The data requirements for solving the expanded KPP-Pos problem are covered in Section 3.2. Section 3.3 presents the expansion of the KPP-Pos measure to include actor and relationship weights. Following that, Section 3.4 develops the use of the p-median to find the key player set that maximizes the modified KPP-Pos measure. Section 3.5 discusses the use of hierarchical clustering to find key player sets. Finally, Section 3.6 covers techniques for applying the modified KPP-Pos measure to social networks with more than one contextual layer.

In Figure 6, the inputs for the WKPP-Pos methodology developed in this study are shown on the left: a social network (single or multi-layered), a list of actor weights, a list of relationship weights and a list of ineligible actors. The social network and the actor and relationship weights are combined to form a weighted social network. This weighted social network and the list of ineligible actors is then used as the input for the p-median or clustering heuristic. The output of which is a set of key players that maximizes the WKPP-Pos measure for the given inputs. The WKPP-Pos measure for each key player set is then calculated and can be used by a decision maker to decided on an influence source of action on the original social network.

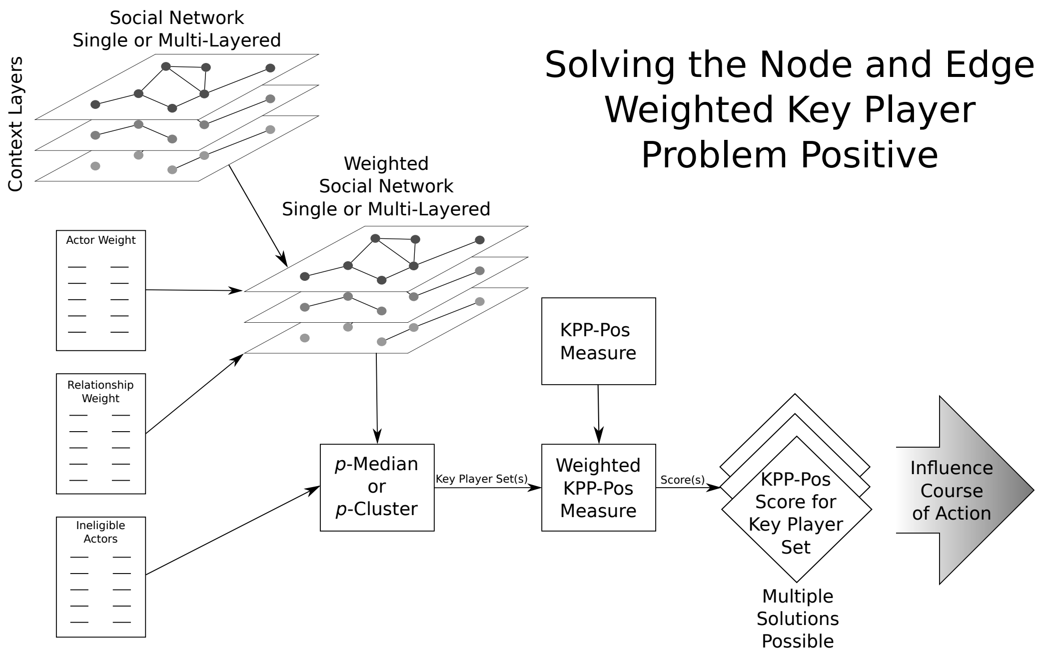

Figure 6: Methodology for solving WKPP-Pos

3.2 Data Requirements

The data that is needed to perform the analysis outlined in this chapter includes the social network, the contexts for the relationships (if they exist), weights for the relationships, weights for the actors, and the number of key players that should be identified. The social network must contain at least one component that has more actors than the desired size of the key player set to be found, $n > k$. Following traditional practice, the largest component of the network will be used to find the key player set. The actor and relationship weights must be real, rational numbers that can be added and multiplied together. The weights of the actors and relationships must meet the assumptions stated in Sections 3.3. This can be accomplished by scaling the weights if required. Additionally, the assumptions stated in Section 2.3 must also hold for the social network.

The source of the actor and relationship weights is not discussed in this study. However, these sources may include another SNA, information gathered about the

actors and relationships in other reports, or from subject matter experts. The restrictions on the weights required by the technique developed in this study are discussed in the following section.

3.3 Expanding KPP-Pos

The definition proposed by Borgatti (2006: p. 29–30) for the KPP-Pos measure does not include relationship or actor weights. In fact, the formulation, as presented in his paper, needs to assume that both are unity, otherwise the formulation is inconsistent. One reason for this assumption of unity is so the measure can be normalized to the range $[0, 1]$ by the method he suggests. However, in this study it is not useful to assume the actors and relationships are unity, but maintaining the normalization is not only useful, but desired.

Hamill (2006: p. 177) briefly describes an approach to add actor and relationship weights to a p-median formulation of Borgatti's KPP-Pos. However, in his formula, the ability to normalize the measure is lost. Additionally, his formula fails to select the correct key player sets under certain conditions. This is illustrated in a simple case of two actors with a relationship as shown in Figure 7. Assume actor 1 is given a weight of 10 and actor 2 is given a weight of 1, where a higher weight means it is more desirable. If the size of the key player set is 1, the obvious choice is actor 1. However, using Hamill's formulation, either actor minimizes the objective function to a value of -11. This is due to the fact that actors in the key player set are treated as if they are a distance of 1 from themselves as assumed by Borgatti (2006: p. 29). However, once weights are added to the actors, this assumption is no longer viable. Further, if the relationship between them is also weighted with a weight of 0.5, then based on Hamill's formulation, actor 2 would be the key player with a value of -21 since actor 1 has a value of -20. This is due to the fact that actor 1 is defined as being 1 unit away from itself while actor 2 is only 0.5 units away from

actor 1. These issues are dealt with in the formulation of the WKPP-Pos presented in this chapter.

Figure 7: Two actor social network

To include relationship weights other than unity in the KPP-Pos measure and maintain the ability to normalize the measure, one restriction needs to be added. The relationship weights must be restricted to the range $[1, \infty)$. This means that the closest any two adjacent actors can be is 1 and the furthest is any real, positive number greater than 1. This preserves the normalization by preventing the individual summation terms in the numerator of Equation 11 from becoming larger than 1. No other changes need to be made to the KPP-Pos measure to allow for the addition of relationship weights.

The addition of actor weights requires further modification beyond the addition of relationship weights. First, the actor weight term needs to be added to the measure. The actor weight, h_j is multiplied by the distance, d_{Kj}, to form a weighted distance from j in the denominator. Second, the same restriction for relationship weights is needed for actor weights. The actor weights are restricted to the range $[1, \infty)$. This allows for the normalization of the measure by again preventing the individual summation terms in the numerator of Equation 11 from becoming larger than 1.

The summation of the inverse of the distance between the key player set, K, and actor j used in Equation 11 is no longer needed. Instead, the summation of the actor weighted distances, $h_j d_{Kj}$, is used. Equation 23 shows the WKPP-Pos measure, ${}^{WD}R$. Equation 24 gives the normalized WKPP-Pos measure, ${}^{WD}R'$. The normalization is accomplished by dividing $n - k$ by the WKPP-Pos measure, where n is the number of actors in the network and k is the designated size of the key player set. This normalization has a range of [0, 1]. The lower bound of zero is reached by having no links between the key player set, K, and the rest of the social network, $V - K$.

$$
\begin{aligned}
{}^{WD}R &= \sum_{j \in V-K} h_j d_{Kj} && (23)\\
{}^{WD}R' &= \frac{n-k}{\sum_{j \in V-K} h_j d_{Kj}} && (24)\\
&\text{for } h_j \in [1, \infty),\ d_{ij} \in [1, \infty)
\end{aligned}
$$

In Equation 23 and 24, h_j is the weight of actor j, d_{Kj} is the minimum distance from any key player to actor j, n is the number of actors in the network, and k is the number of key players. This formulation uses the distance $d_{ii} = 0$ for the distance between an actor and itself. For this reason, the normalizing factor used in Equation 24 is $n - k$, rather than n as Borgatti proposes.

The results of the normalized WKPP-Pos for a given key player set is a value between 0 and 1. The inverse of that value is the average of number relationships an actor is from the nearest key player. For instance, if the normalized WKPP-POS measure for a given key player set was 0.5, then the average distance between the actors in the network and their closest key player is 2.

The Python source code for the WKPP-Pos formulation used in this study is given in Appendix A. This code uses the output from the p-median or hierarchical

clustering programs, discussed in the next two sections, to calculate the WKPP-Pos for a given set of key players.

There may be cases when it is known that some actors are not eligible to be members of the selected key player set in the network. While a variation from Borgatti's definition of a key player, this may occur due to operational requirements or restrictions. For example, this might occur due to limited access to those actors or for other reasons such as political or safety concerns. These actors can still pass information or goods in the social network, so removing them completely is not an option. Instead, they are removed from the list of potential key players during the analysis. In these cases, the ineligible actors need to be identified and handled properly in the p-median and hierarchical cluster analysis. The number of eligible actors still needs to be larger than the key player set size that is desired. The technique for handling ineligible actors for the p-median and hierarchical clustering formulations are discussed in each of the following sections.

Social networks are dynamic and change over time. Actors enter and leave the network and relationships are formed and broken. As this happens, having a measure that is normalized allows the analyst to track the changes to the effectiveness of the key player set that was previously selected. Additionally, historic data about operations that used key player sets can be used to look at the correlation between the success of the operation and the KPP-Pos score for that operation. This type of analysis could lead to a standard WKPP-Pos score to use in planning operations to improve the success of those operations.

3.4 p-median

As Hamill (2006: p. 176) notes, the p-median finds the optimal solution of the KPP-Pos measure by minimizing the summation of distances in the denominator in Equation 11 for a given size, p, of the key player set. If actor weights are used, then a weighted p-median problem is used to find the optimal key player set. The p-median

formulation used in this study is shown in Equations 25–29 (Marianov and Serra, 2009: p. 4). This formulation is a minimization of the denominator in Equation 24.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{M} h_i d_{ij} x_{ij} \tag{25}$$

$$\text{subject to:} \sum_{j=1}^{M} x_{ij} = 1 \quad i = 1, \ldots, N \tag{26}$$

$$\sum_{j=1}^{M} y_j = p \tag{27}$$

$$x_{ij} \leq y_j \quad i = 1, \ldots, N,\ j = 1, \ldots, M \tag{28}$$

$$x_{ij}, y_j \in 0, 1 \quad i = 1, \ldots, N,\ j = 1, \ldots, M \tag{29}$$

To deal with actors that are ineligible to be selected for the key player set, the y_i term for each of the ineligible actors is fixed at zero. This causes the solution to the p-median to exclude the ineligible actors from the key player set. Performing the analysis with and without this restriction will quantify the reduction in the key player set score due to the ineligibility of a set of actors. This could also be done be executing the analysis with and without a single member of the ineligible set. This would develop a penalty for excluding an individual key player.

The python source code for the p-median formulation used in this study is given in Appendix A. This code calls lp_solve, an open-source command-line based mixed integer linear programming solver, to solve the p-median problem. The code returns the p actors which are the medians for the given weighted social network and the actors assigned to each of the p medians.

Using the p-median to solve for one key player in the simple example given in Figure 7 where actor 1 has a weight of 10 and actor 2 has a weight of 1, results in a key player set of $\{1\}$ and a normalized weighted KPP-Pos value of 1.0. Using the

same actor weights and adding an relationship weight of 2 between actors 1 and 2, the same key player set is found with a weighted KPP-Pos value of 0.5.

3.5 Hierarchical Clustering

Hierarchical clustering can be used in place of p-medians to find a key player set. It is not guaranteed to find the optimal solutionto the model given in Equations 25–29, but it can serve as a heuristic for large networks. The quality of hierarchical clustering as a heuristic for the WKPP-Pos is only addressed for the case studies that are presented in this study. The distance matrix that is passed to the algorithm is first computed using the relationship weights. The method of computing distances between clusters used in this formulation is averaging; however, other methods exists, such as furtherest point, closest point, and centroid. In addition, different measures for distance can be used. This study uses the length of the path between actors as the distance measure. Once the clustering has occurred, the 1-median of each of the k clusters is calculated using the p-median formulation in Equations 25–29. This results in a set of k key players.

To handle ineligible actors, the list of actors that are ineligible are used when determining the 1-median of each cluster as described in Section 3.4. This will result in the same key player set as the p-median technique if the network is clustered in the same way as in the p-median.

The python source code for the hierarchical clustering formulation used in this study is given in Appendix A. The clustering algorithm used in this code is from the Python package SciPy (Jones *et al.*, 2001). The code returns the p actors which are the medians for each of the p clusters formed by the hierarchical clustering routine and the actors assigned to each of the p medians.

Using hierarchical clustering to solve for one key player in the simple example given in Figure 7 where actor 1 has a weight of 10 and actor 2 has a weight of 1, results in a key player set is $\{1\}$ and a normalized weighted KPP-Pos value of 1.0.

Using the same actor weights and adding the an relationship weight of 2 between actors 1 and 2, the same key player set is found with a weighted KPP-Pos value of 0.5. This sample example demonstrates a case when the hierarchical clustering and the p-median, example in Section 3.4, give the same results.

One down fall of hierarchical clustering technique is that there is no guarantee that the clusters that are formed will be connected. This could yield a cluster in which the 1-median can not reach all actors, resulting in an infinite distance from the median to the disconnected actors. The easiest solution to this problem is to reassign the disconnected actors to a more appropriate cluster. This problem was not seen in the datasets used in Chapter IV with the averaging distance method used for clustering.

3.6 Multiple Layers

When dealing with multiple contextual layers, each of l layers can be treated as a separate social network and key player sets found for each. Then the social network can be combined into a single layer and the key player set found for that social network. Any actors that overlap in sets are good candidates for the final key player set. A sub-optimal key player set may be selected in a layer due to that set being an optimal set in another layer or in the combined social network. The KPP-Pos value can be used to determine the reduction in optimality due to selecting a sub-optimal key player set in a layer. Calculating a criticality index (the percentage of time the actor was in any of the $l + 1$ key player sets) can aid in identifying individuals who are key in a number of contexts.

This chapter developed the WKPP-Pos measure that allows for the inclusion of actor and relationship weights. Restrictions for the values of the actor and relationships were identified so that the newly developed measure can be normalized. Normalization of the measure is important for comparison as a social network changes over time and for comparison between different social networks. Techniques were de-

veloped for using the p-median to find optimal solutions to the WKPP-Pos measure and for using hierarchical clustering as a heuristic for finding solutions to the WKPP-Pos measure. Finally, a technique for finding key player sets for multi-layered social networks was developed. In the next chapter, these techniques are exercised on a number of sample social networks.

IV. Case Study Results

4.1 Introduction

In this chapter, a series of case studies are used to show a proof of concept for the weighted key player problem positive, WKPP-Pos, that is developed in Chapter III. Section 4.2 discusses the analysis of five case studies using the WKPP-Pos measure to find optimal key player sets. Section 4.3 discusses the performance of the p-median technique and the hierarchical clustering technique in analyzing the case studies in this study.

4.2 Case Studies

The following case studies, taken from open literature, demonstrate the utility of incorporating additional information about actors and relationships in the KPP-Pos. When possible, actual data about actors and relationships is used to generate the actor and relationship weights. However, some of the following case studies did not have sufficient data on the actors of relationships to develop weights. In those cases no weights are applied, weights were generated based on other data that was available or random weights were generated. All scores for the datasets have been rounded to 5 decimal places.

4.2.1 Method's Camp Dataset. The following data set is provided with Analytic Technologies' Key Player software package (Ana, 2003). The data was collected by asking each of the 18 attendees of a camp program to rank their interactions with the other attendees using ordinal ranking from the most interaction, 1, to least, 17. The social network data provided in the software package contains only the relationships ranked 1, 2 and 3 by each actor. In this case study, the question of how node weights effect the optimal key player set is investigated. All relationships have a weight of 1 in this demonstration since they represent the highest interactions for each actor.

A network analysis of the full social network, incorporating all of the relationship data, was previously performed and the actors that appeared most frequently in the top three of all the measures in that analysis are actors 1, 2, 3, 10, 16 and 17 (CASOS, 2008). Actors 1, 2, and 3 were in the top three approximately 80% of the time and actors 10, 16 and 17 were in the top three approximately 20% of the time. The percentages are used as weights for the actors in this analysis. Actors 1, 2 and 3 are given a weight of 8 and actors 10, 16 and 17 are given a weight of 2. All other actors were given a weight of 1. This weights are only intended to serve as example weights and do not have have meaning beyond this example. Figure 8 depicts the social network with the actor size scaled by their weight.

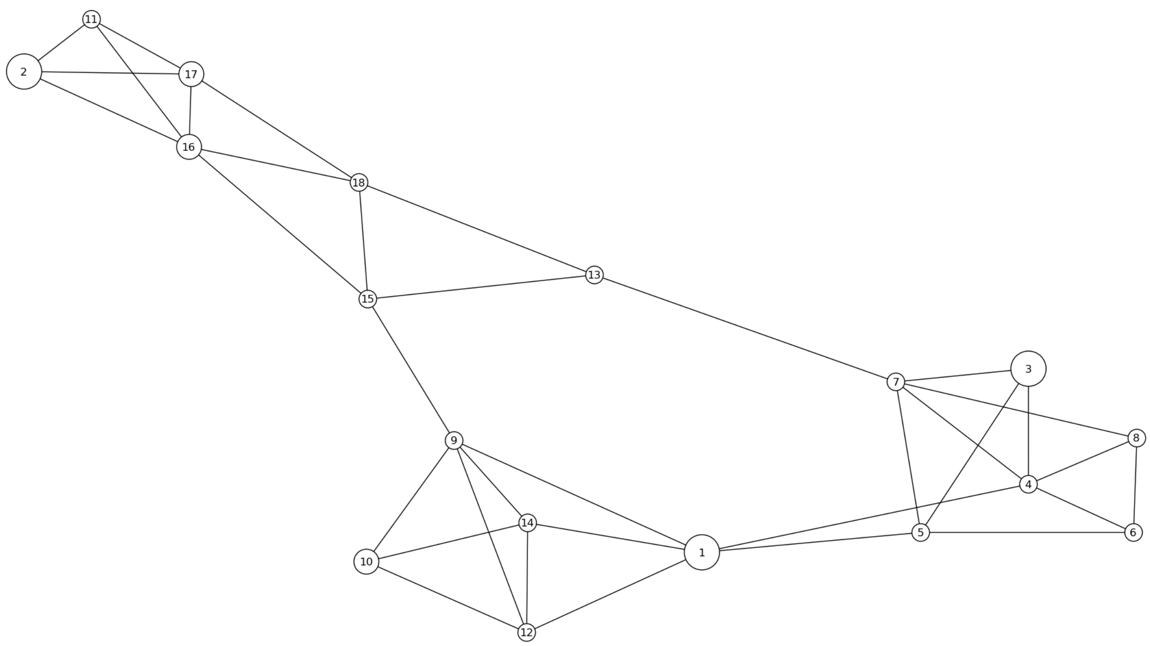

Figure 8: Method's Camp social network with actors scaled by weight

The analysis of the key players was performed with and without ineligible actors. Three of the top six actors were chosen at random to become ineligible actors for this analysis. The ineligible actors for this analysis are 1, 3, 10 and 16. The use of the ineligible actors is to demonstrate the effect of not being able to choose higher weighted actors for the key player set. Comparing the measure with and without

ineligible actors quantifies the reduction in the effectiveness of the restricted key player set and can identify actors that are critical to an effective key player set.

The results of performing a key player analysis of this dataset is presented in Table 10. KPP-Pos is the result of processing the dataset with an unweighted p-median technique. The WKPP-Pos is the result of processing the data with the p-median technique shown in Equations 25–29. The cluster results are from processing the data with the hierarchical clustering technique described in Section 3.5. Finally, each of these techniques is run again with the ineligible actor set. The unweighted scores are the results of calculating the normalized WKPP-Pos, ${}^{WD}R'$, without using the actor weights. The weighted scores are the results of calculating the normalized WKPP-Pos, ${}^{WD}R'$, using the actor weights. This allows for comparison of how actor weights impact the selecting and scoring of key player sets. The same terminology is used for all the following datasets.

Table 10: Results for Methods Camp Data Set

	Key Player Set	Unweighted Score	Weighted Score
KPP-Pos	{4, 7, 9, 17}	1.00000	0.37838
WKPP-Pos	{1, 2, 3, 10}	0.73684	0.66667
Cluster	{1, 2, 3, 15}	0.77778	0.63636
KPP-Pos Ineligible Actors	{4, 7, 9, 17}	1.00000	0.37838
WKPP-Pos Ineligible Actors	{2, 7, 9, 17}	0.93333	0.45161
Cluster Ineligible Actors	{2, 4, 9, 15}	0.87500	0.42424

From the results in Table 10, it can be seen that if the network is not weighted, the p-median technique finds the maximum optimal key player positive set as defined by Borgatti. The key player set {4, 7, 9, 17} is able to reach all other actors using only one relationship. Since this set does not contain any of the ineligible actors, for this example, it is also the maximum optimal key player set for the case which includes ineligible actors. Figure 9 depicts the optimal key player set when the actor weights are not included in the p-median problem.

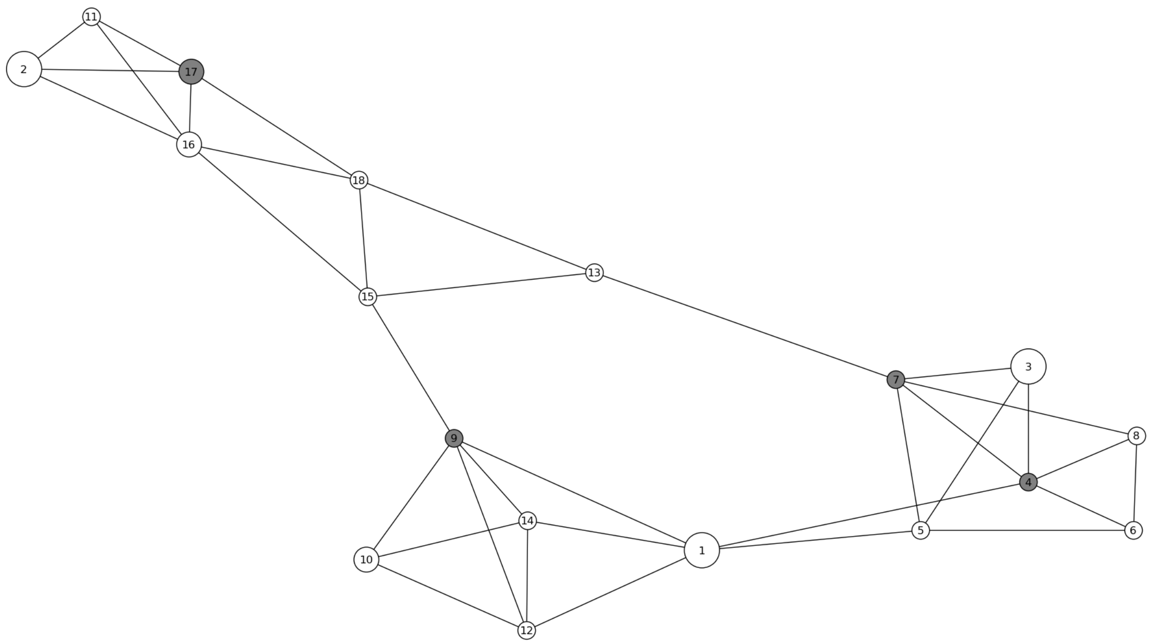

Figure 9: Method's Camp social network with the unweighted key player set highlighted in grey

However, once the weights of the nodes are taken into account in the measure, the key player set {6, 7, 9, 16} does not perform as well. Instead, the key player set, {1, 2, 3, 10}, shown in Figure 10, found using the weighted p-median technique, has a higher normalized WKPP-Pos score. This means that the key player set found using the weighted p-median contains actors that have higher weights or actors that are closer to other actors which are weighted higher than compared to the standard key player set. This is seen in Figure 10 as the three highest weighted actors, 1, 2 and 3 are selected as well as actor 10 which is not adjacent to any other highly weighted actor.

Finally, when ineligible actors are considered, the weighted p-median technique finds the key player set {2, 7, 9, 17}, shown in Figure 11, which also has a higher normalized WKPP-Pos score than the key player set found using the unweighted p-median technique. This demonstrates that if the objective is to select actors based on their characteristics as well as their structural position, a technique that incorporates actor weights is desirable. Without incorporating information about actor

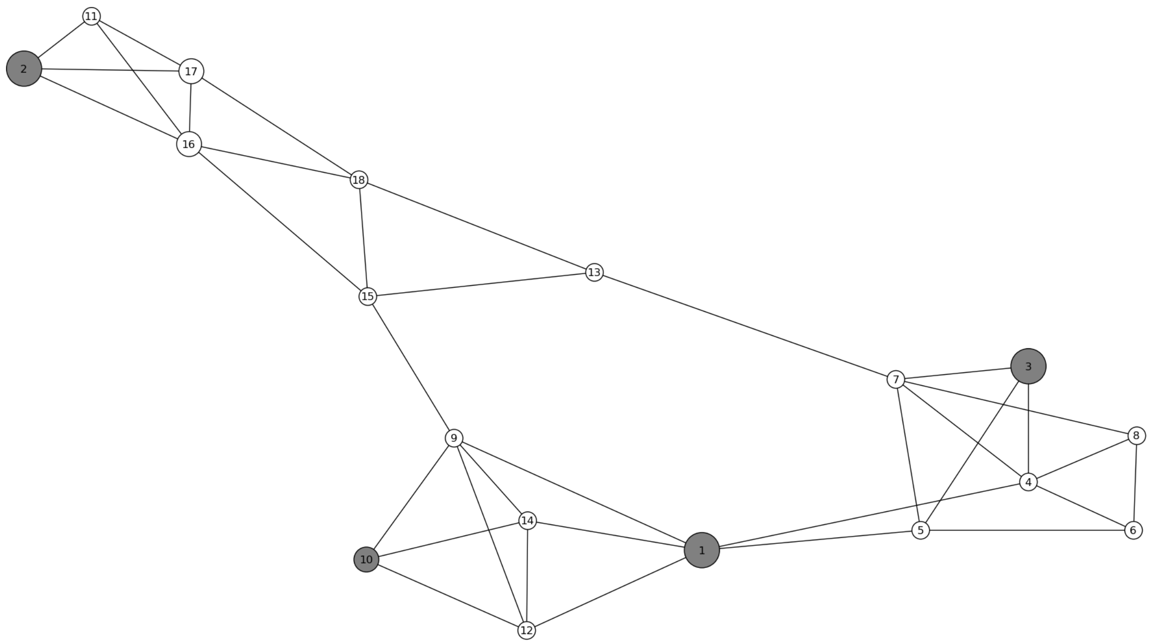

Figure 10: Method's Camp social network with the weighted key player set highlighted in grey

characteristics in the key player analysis, a suboptimal key player set would have been selected in this example. The suboptimal key player set would have the potential to decrease the effectiveness of any operation being performed against this social network.

4.2.2 Hartford Drug User Dataset. The Hartford Drug User data set was collected to study and reduce the spread of HIV between drug users in the city of Hartford, Connecticut (Weeks *et al.*, 2002). The full dataset consists of one large component of 193 actors and a number of smaller components with fewer than 4 actors each. The larger component of 193 actors is the focus of this analysis and has been referred to as the network in this analysis. The smaller components have been removed from the dataset because they are not connected to the larger component and hence the risk of HIV being spread to them is assumed to be zero. The network considered consists of 193 actors and 273 relationships. In this analysis, the goal is to identify a group of actors that can quickly spread information about proper handling

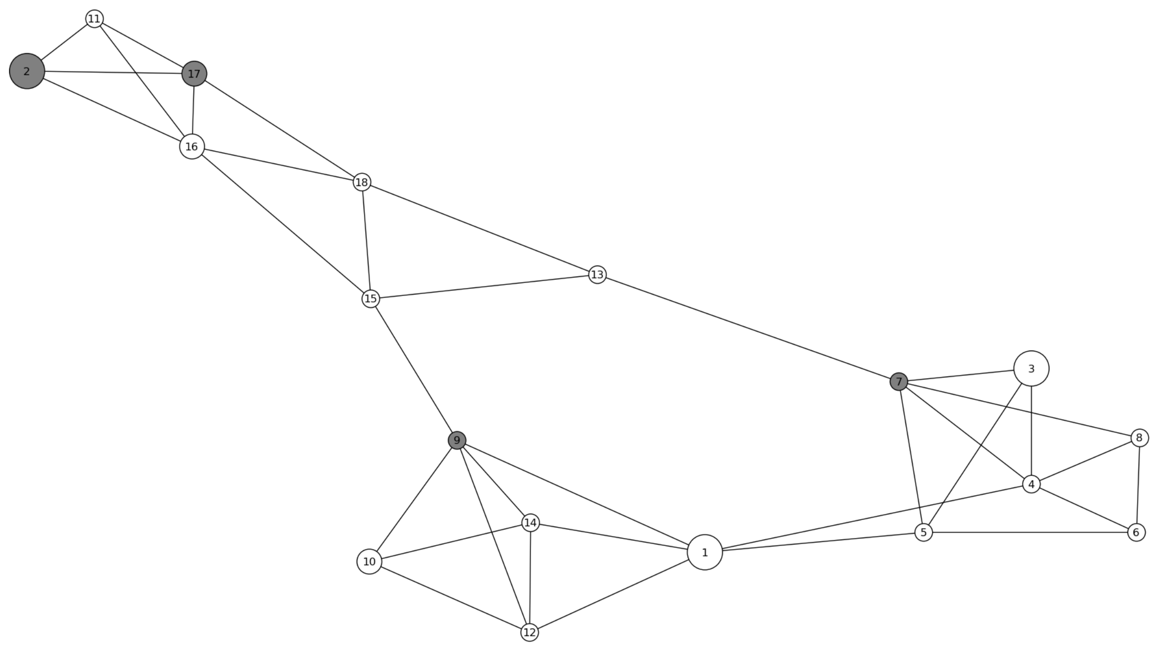

Figure 11: Method's Camp social network with the weighted key player set highlighted in grey, accounting for ineligible actors 1, 3, 10, 16

and cleaning of needles used to inject drugs in an attempt to slow the spread of HIV within the network. A key player set of 10 was selected based on approximately 5% of the network size. In this dataset, the effects of using edge weights on the key player set are examined.

There was no data available that could be used to weight the relationships in the network, so the weights for the relationships were assigned integer values uniformly distributed on the range [1, 10]. A weight of 1 implies that the actors have a very close relationship while a weight of 10 implies a very weak relationship. Figure 12 depicts the social network for this dataset with the edge thickness denoting the strength of the relationship with thicker being stronger. Data that could be used to weight the edges might include the number of shared needles between actors, the number of times arrested together or from surveys of the groups, if they would cooperate in truthfully filling out the surveys.

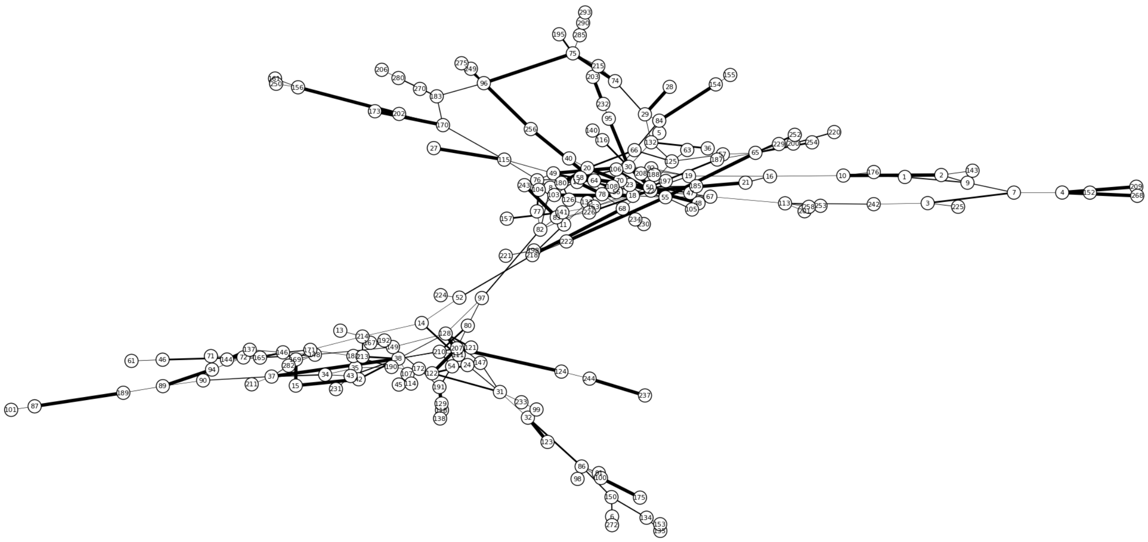

Figure 12: Hartford Drug User social network with edges thickness denoting relationship strength

The ineligible actors for this analysis are 7, 38, 50, 75, 86 and 120. These were selected at random from the set of actors that were members of the key player set for the first three analyses: KPP-Pos, WKPP-Pos and cluster. Ideally, the ineligible actors would include actors that are known not to support or trust public programs. This might include known drug dealers if they feared their creditability or business might be compromised by taking part in the program.

The key players identified by each of the techniques are shown in Table 11. All

Table 11: Key Player Sets for Hartford Drug User Data Set

	Key Player Set
KPP-Pos	{7, 30, 38, 50, 75, 86, 89, 170, 171, 191}
WKPP-Pos	{7, 38, 50, 66, 86, 94, 96, 113, 191, 218}
Cluster	{7, 38, 50, 86, 94, 99, 113, 155, 156, 290}
KPP-Pos Ineligible Actors	{9, 30, 37, 55, 64, 94, 122, 150, 181, 191}
WKPP-Pos Ineligible Actors	{9, 20, 30, 37, 55, 94, 96, 113, 150, 191}
Cluster Ineligible Actors	{9, 30, 37, 94, 99, 113, 150, 155, 156, 290}

three techniques had similar key player sets, which is reflected in the scores for the key player sets shown in Table 12. The key player set for the WKPP-Pos makes use

Table 12: Scores for Hartford Drug User Data Set

	Unweighted Score	Weighted Score
KPP-Pos	0.53353	0.10952
WKPP-Pos	0.46447	0.12906
Cluster	0.42166	0.11193
KPP-Pos Ineligible Actors	0.52890	0.11546
WKPP-Pos Ineligible Actors	0.43468	0.12543
Cluster Ineligible Actors	0.37500	0.10578

of the stronger relationships in the network to bridge between the various sections of the social network. Many of the key players selected have strong relationships with other actors in the network, allowing them to be as close to as many actors as possible. This is seen with actors 38, 94, 96, and 218, all of which have at least one strong connection incident to them.

Based on the these results, it can be seen that including edge weights can have a discernible impact on the scores of a key player set. The optimal 10 actor key player set for the weight network is {7, 38, 50, 66, 86, 94, 96, 113, 191, 218} and only scores 0.12906. In the unweighted network, the optimal key player set scored over 4 times higher with a score of 0.53353. The weighted key player set is shown in Figure 13 with the key players highlighted in grey. It can be seen that the key player set uses the stronger relationships, the thicker lines, to increase their reach across the network. The score for the optimal weighted key player set decreases to 0.12543 when ineligible actors are included in the analysis. This key player set is shown in Figure 14 with the key player set highlighted in light grey and the ineligible actors highlighted in dark grey. The biggest change in the network is seen in the dense upper section of the network. Previously, two key players covered this group of actors, but now three are required to cover it.

To achieve a higher score, meaning the key player set is closer to the rest of the actors and able to influence them more directly, the key player size would need to be increased. Increasing the size of the key player set allows more of the actors to be

directly influenced by the key player set and reduce the path distance to nonadjacent actors. In terms of reducing the spread of HIV within this group, a key player set of 10 is an average distance of 7.7483 from all other actors in the network. This distance is likely too high to effectively spread information about sterilizing needles to a large percentage of the network. The program would need to increase the size of the key player set to reduce the average distance between the key player set and the other actors to effectively spread information in this network.

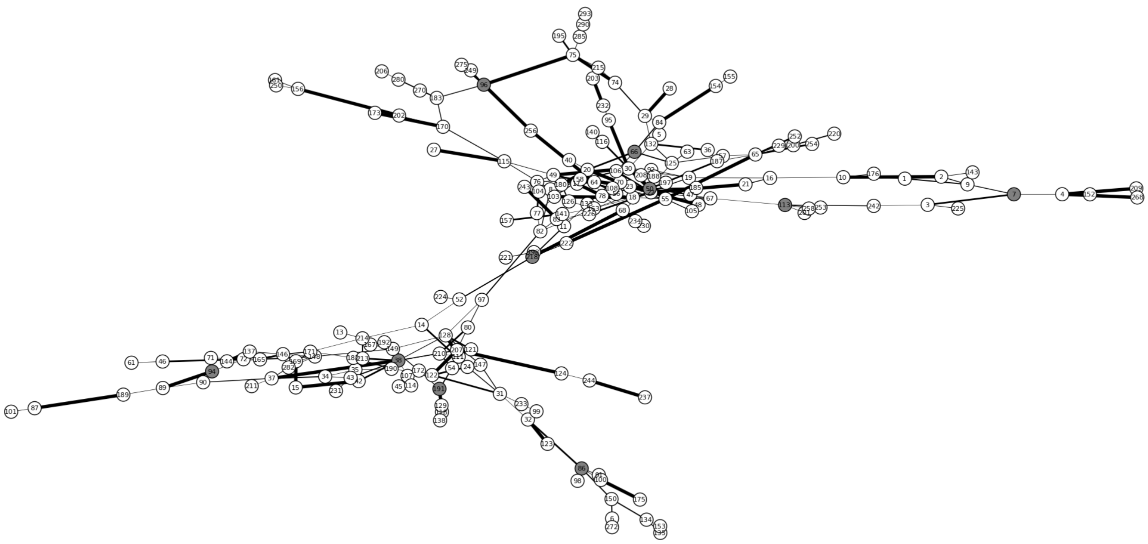

Figure 13: Hartford Drug User social network with the weighted key player set highlighted in grey

4.2.3 Krebs' 9/11 Hijackers Trusted Prior Contacts Dataset. The following data set was compiled by Krebs using open source literature about the 9/11 hijackers (Krebs, 2002). This dataset shows the trusted prior contacts between the 19 actual hijackers from the four flights on 9/11. In this analysis the goal is to identify a set of four key players that might have been influenced to provide data about the attacks or could have been placed under observation in the hopes of gathering information about the attacks. The choice of four actors for the key player set size in this analysis is two fold. First, there were four hijacking teams that information would have to have been gathered on to determine each team's mission. Second, the number of

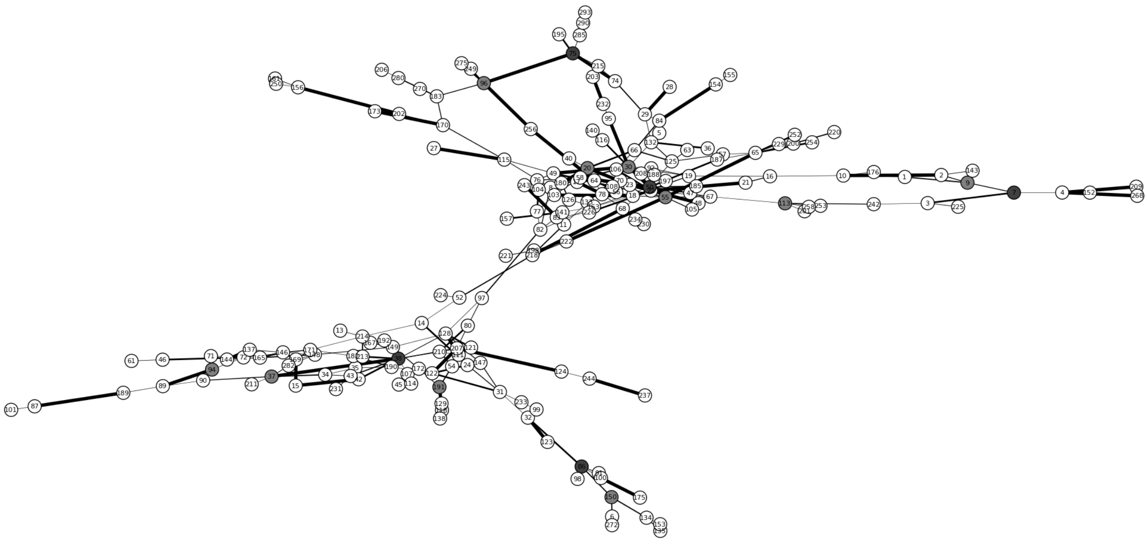

Figure 14: Hartford Drug User social network with the weighted key player set highlighted in light grey and ineligible actors highlighted in dark grey

actors that can be influenced or observed cannot be a large percentage of the actual network, so as not to draw the attention of the actors in being observed.

The ineligible actors for this analysis are 5, 6, 7 and 18. These nodes represent the leaders of each of the four hijacking groups. The leaders were given an actor weight of 10 and all other actors were given an actor weight of 1. Edge weights different than one were not assigned for this analysis. The social network is shown in Figure 15 with actors size relating to their weight. The hijacking teams are:

- AA #11: Actors 1, 2, 3, 4, 5
- UA #93: Actors 7, 10, 13, 14
- UA #175: Actors 6, 8, 9, 11, 12
- AA #77: Actors 15, 16, 17, 18, 19

The results of the key player analysis is shown in Table 13. With a key player set size of four, there are multiple optimal solutions to the unweighted network problem. This can be seen by looking at the unweighted scores for the KPP-Pos and the WKPP-Pos; both of which scored 0.93750 with a different key player set. Figure

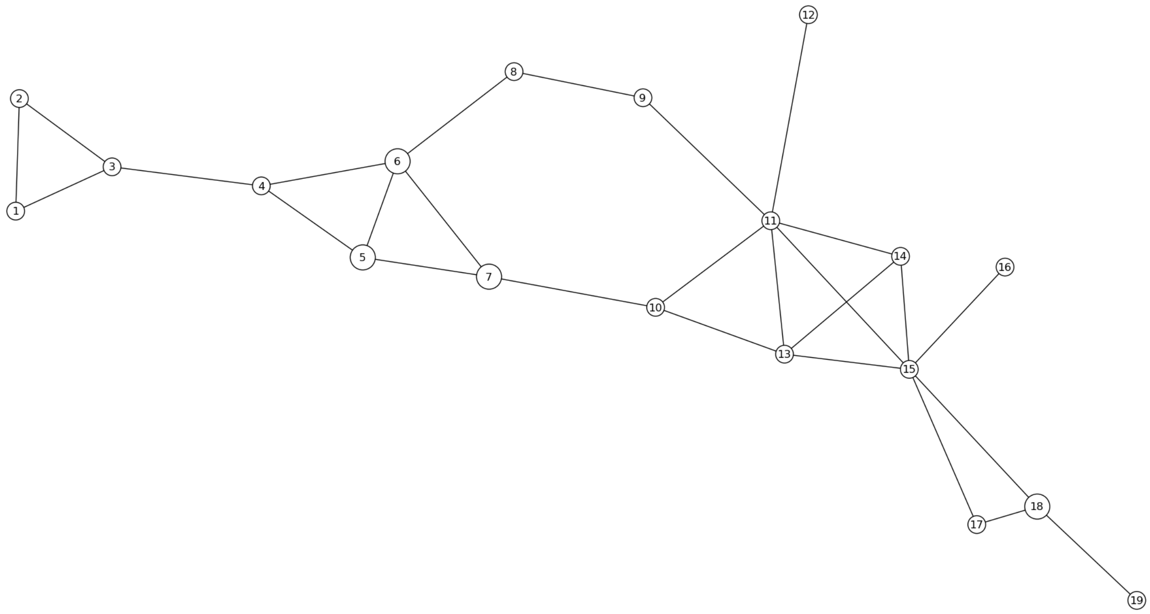

Figure 15: Krebs' 9/11 Trusted Prior Contacts social network with actor size representing actor weight

16 depicts the key player set that was found without consideration for actor weights and Figure 17 depicts the key player set that was found when actor weights were considered. Once actor weights and ineligible actors are included in the analysis, the similarities between the sets disappear.

Table 13: Results for Krebs' 9/11 Hijackers Trusted Prior Contacts Data Set

	Key Player Set	Unweighted Score	Weighted Score
KPP-Pos	{1, 6, 11, 15}	0.93750	0.53571
WKPP-Pos	{3, 6, 11, 18}	0.93750	0.62500
Cluster	{3, 6, 12, 18}	0.71429	0.51724
KPP-Pos Ineligible Actors	{2, 4, 11, 15}	0.83333	0.39474
WKPP-Pos Ineligible Actors	{4, 10, 11, 15}	0.78947	0.42857
Cluster Ineligible Actors	{3, 4, 12, 15}	0.68182	0.35714

This example shows that actor characteristics can be used as a possible discriminator for multiple optimal solutions. In this example, either of the key player sets found using the unweighted and weighted p-median will be mathematically optimal

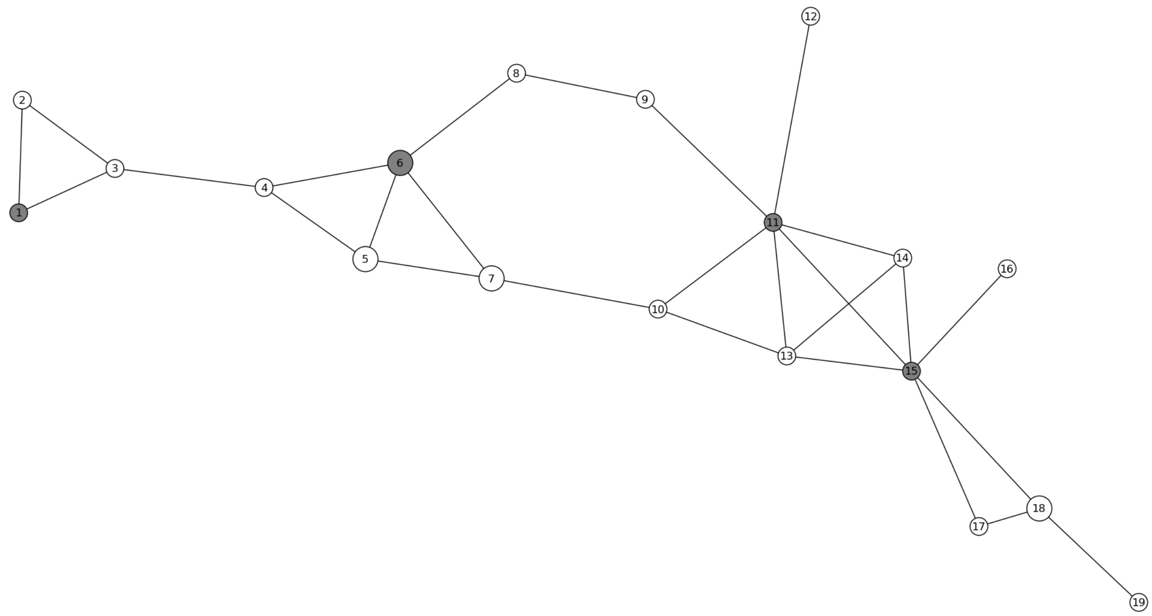

Figure 16: Krebs' 9/11 Trusted Prior Contacts social network with the unweighted key player set highlighted in grey

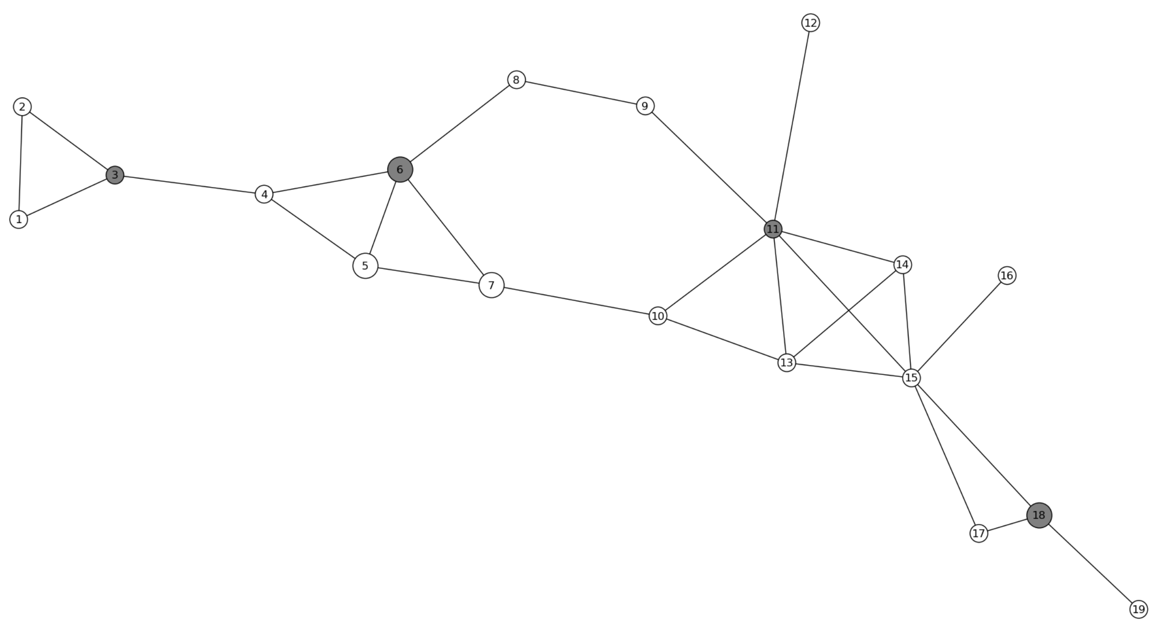

Figure 17: Krebs' 9/11 Trusted Prior Contacts social network with the weighted key player set highlighted in grey

for the unweighted network. However, if information about the actors is available, it can be used to select a key player set that is optimal in both cases.

4.2.4 Krebs' Full 9/11 Hijacker Dataset. The following data set was compiled by Krebs using open source literature about the 9/11 terrorist (Krebs, 2002). This dataset contains all contacts between individuals associated with the terrorist attacks on 9/11. The dataset contains 63 actors and 154 relationships. In this analysis, the difference between Borgatti's key player set and the two key player sets found using the p-median and the hierarchical clustering techniques are compared. No actor or relationship weights were used for this analysis so Borgatti's results could be directly compared. The social network is shown in Figure 18.

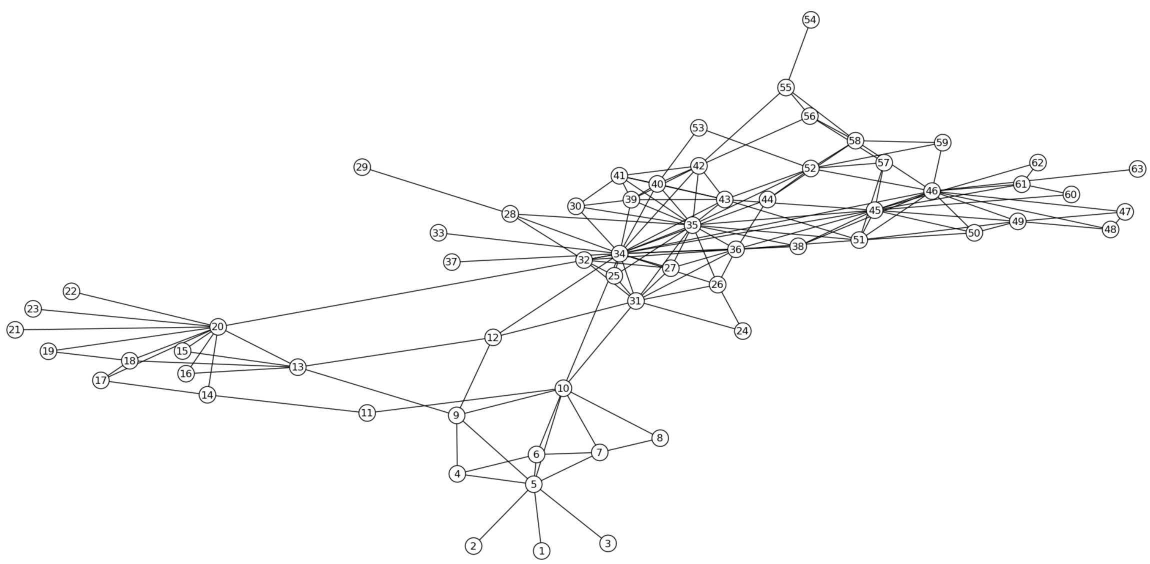

Figure 18: Krebs' Full 9/11 Hijacker social network

Borgatti used a heuristic to find a set of 3 actors that cover the network in two links or less (Borgatti, 2006: p. 31). Using the p-median to solve for a key player set of size 3 results the same key player set obtained by Borgatti as seen in Table 14 and depicted in Figure 19. As Borgatti was using a heuristic, he did not claim to have found the optimal solution, however in this example he did. Using the hierarchical clustering technique, a key player set is found that is not optimal and the key player

set does not meet Borgatti's requirement that all actors be within two relationships of a member of the key player set.

Table 14: Results for Krebs' Full 9/11 Hijackers Data Set

	Key Player Set	Score
Borgatti's Key Player Set	{5, 34, 46}	0.72289
WKPP-Pos	{5, 34, 46}	0.72289
Cluster	{5, 20, 34}	0.70588

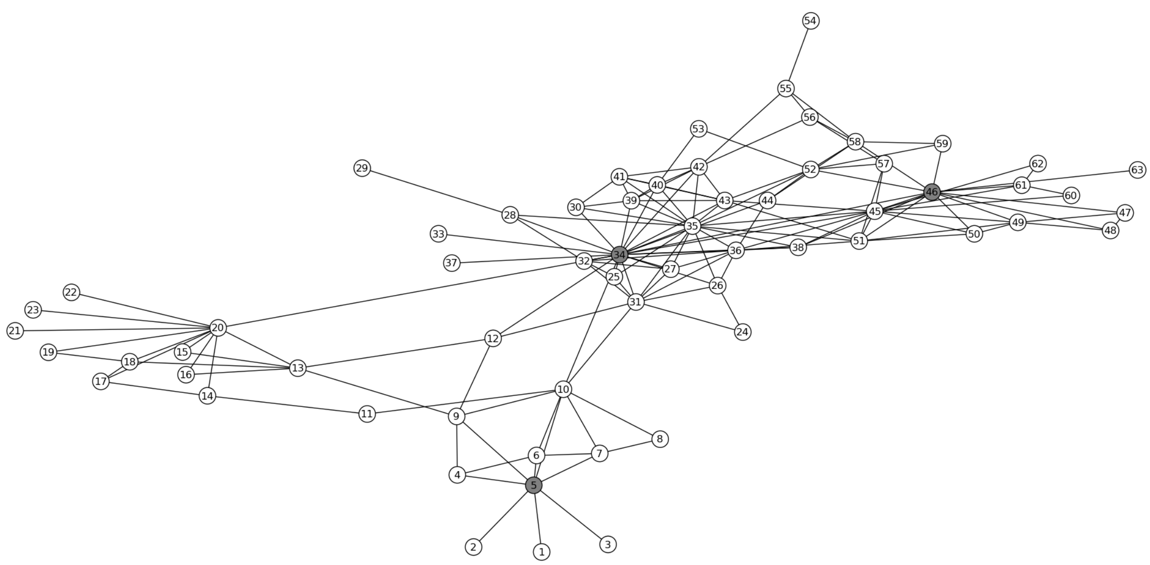

Figure 19: Krebs' 9/11 Full Hijacker social network with the optimal key player set highlighted in grey

The results of this analysis show that a key player set of only 3 actors is able to reach all 63 actors in this network using only two relationships. Further, based on the measure of 0.72289, the average distance from the key player set to any other actor in the network is 1.3833. This key player set would have been the optimal set of actors to place under observation to gather information about the plans of this group. With actor and relationship weights, a different set might have been found to be a better key player set for observation.

4.2.5 Krackhardt High-tech Managers Dataset. The following dataset was compiled by Krackhardt on a group of managers at a company that manufactured high-tech machinery (Krackhardt, 1987: p. 118). Krackhardt collected relationship data for three different contexts: advice seeking, friendship, and job structure. This results in a super-sociomatrix for the 21 managers in this social network. The social networks are converted to undirected graphs for this analysis as the KPP-Pos has not been extended to include directed graphs. The three layers of this social network are shown in Figures 20, 21 and 22. The objective in this analysis is to identify a set of key players that scores high across the three context layers. This set of actors is well suited to spread information throughout the network and also gather information in the network. For this analysis, the size of the key player set is 3.

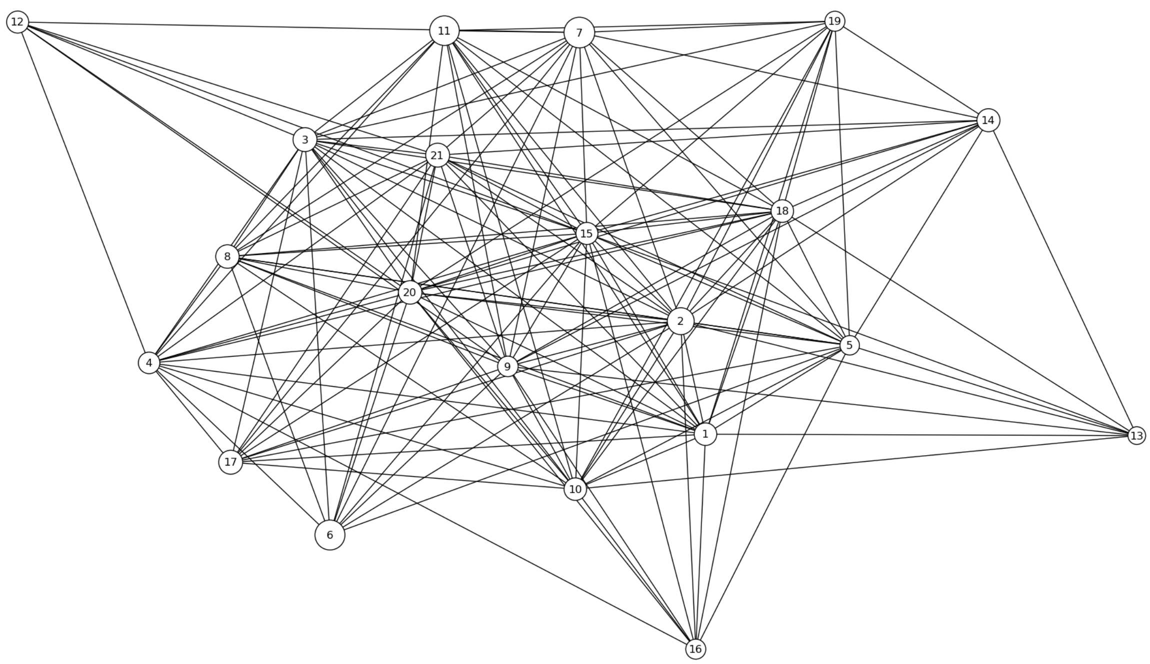

Figure 20: Krackhardt High-tech Managers Advice seeking network

The weight of an actor can vary between contexts depending on the characteristics of that actor and the context being analyzed. For this analysis, the top level managers were ranked higher in the job structure context compared to the lower level managers. While in the advice seeking context the number of years in the industry

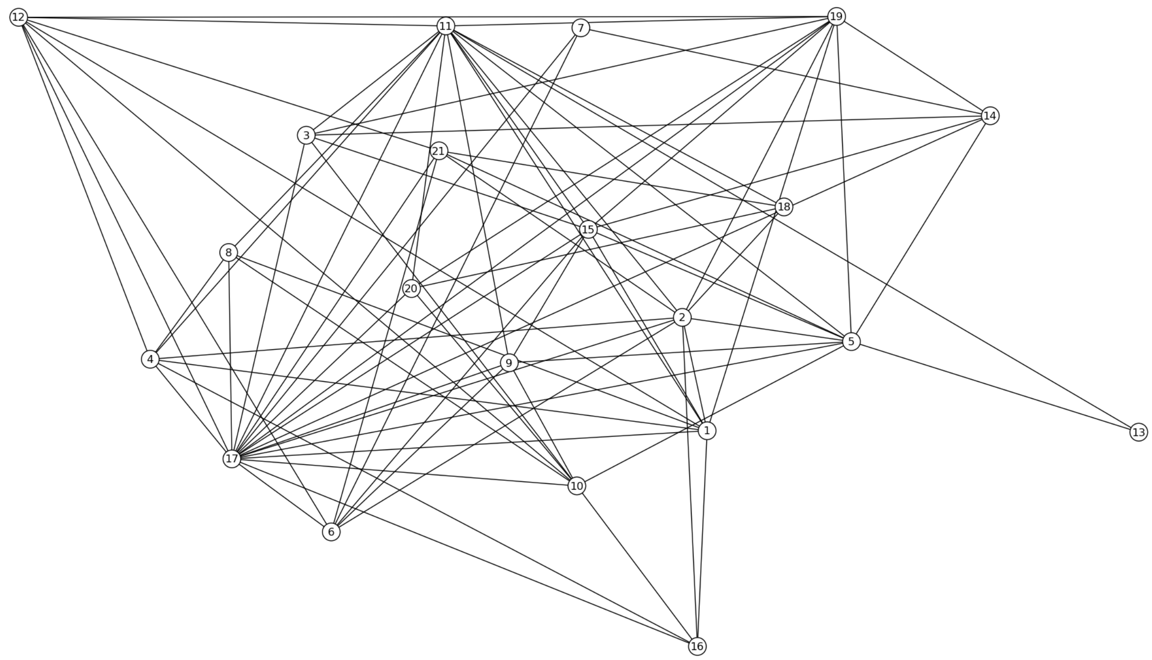

Figure 21: Krackhardt High-tech Managers Friendship network

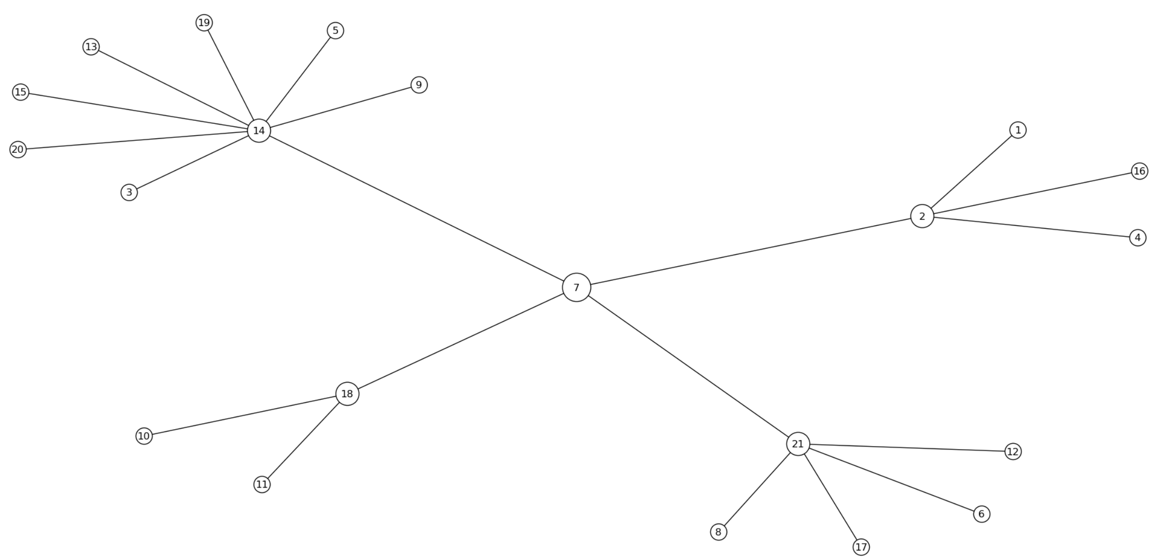

Figure 22: Krackhardt High-tech Managers Job structure network

was used to compute the node weights by rounding up the number of years. The friendship context is left unweighted in this analysis. The weights for each actor in each context is shown in Table 15.

Table 15: Node Weights by Context for Krackhardt's Dataset

Actor	Advice Seeking	Friendship	Job Structure
1	10	1	1
2	20	1	5
3	13	1	1
4	8	1	1
5	4	1	1
6	28	1	1
7	30	1	10
8	12	1	1
9	6	1	1
10	10	1	1
11	27	1	1
12	9	1	1
13	1	1	1
14	11	1	5
15	9	1	1
16	5	1	1
17	13	1	1
18	10	1	5
19	5	1	1
20	12	1	1
21	13	1	5

To begin with, a key player set is found for each context layer. Table 16 shows the key player set for each layer and the resulting WKPP-Pos score. Actors 7 and 11 each show up twice in the set of three key player sets. Actor 7 is in the advice seeking key player set and the job structure key player set. Actor 11 is in the advice seeking key player set and the friendship key player set.The next step is to combined the 3 context layers into a single social network. The relationships from all three contexts are placed into a single social network and any duplicate relationships are removed. The weights for the actors were the summation of their weights in all three

contexts. The key player set found using the p-median for this network is shown in the Combined Network row of Table 16.

Table 16: Key player sets by context in Krackhardt's dataset

Context	Key Player Set	WKPP-Pos
Advice Seeking	{6, 7, 11}	0.10169
Friendship	{2, 11, 17}	1.00000
Job Structure	{7, 14, 21}	0.58065
Combined Network	{2, 6, 7}	0.07965

In the combined network, actors 2, 6 and 7 are members of the key player set that provides the optimal WKPP-Pos score. Actors 2, 6, 7 and 11 all show up at least twice between the three context layers and the combined network. Three of these form the start of a multi-layered key player set. Each combination is analyzed to see which scores highest. The results of each combination is shown in Table 17. Using the average score as the deciding factor, the key player set of {2, 7, 11} would be chosen for this social network. This key player set is shown in Figures 23, 24 and 25 with the key player set highlighted in grey.

The key player set, {2, 7, 11}, contains the top manager, actor 7, the middle manager with the most years of experience, actor 2, and the low level manager with the second most years of experience, actor 11. Actor 6 is the low level manager with the most years of experience and was one of the final 4 actors in consideration. If it was known that one of the layers were more important than the others for the spreading or gathering of information, a weighted average could be used to determine the key player set.

The ability to identify a key player set in a multiple layered social network allows an analyst to be selective in which relationship context or contexts a key player set should be optimal. This can be used to insure information is spread or gathered from a subset of actors with minimal information leakage outside the contexts of interest.

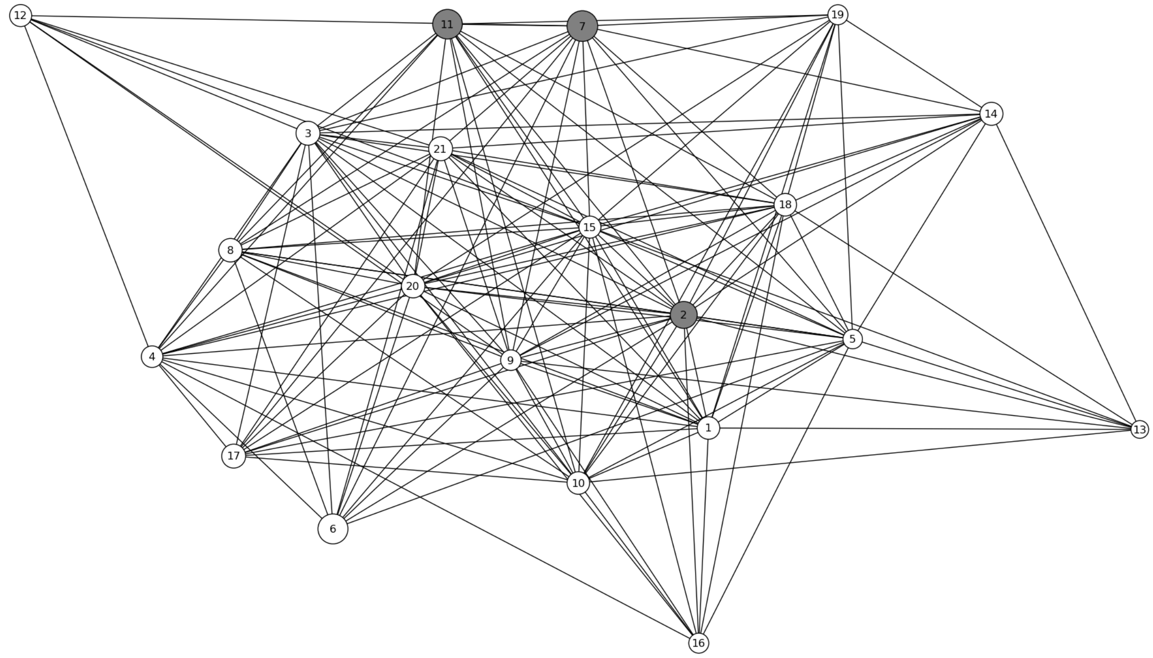

Figure 23: Krackhardt High-tech Managers Advice seeking network with final key player set highlighted in grey

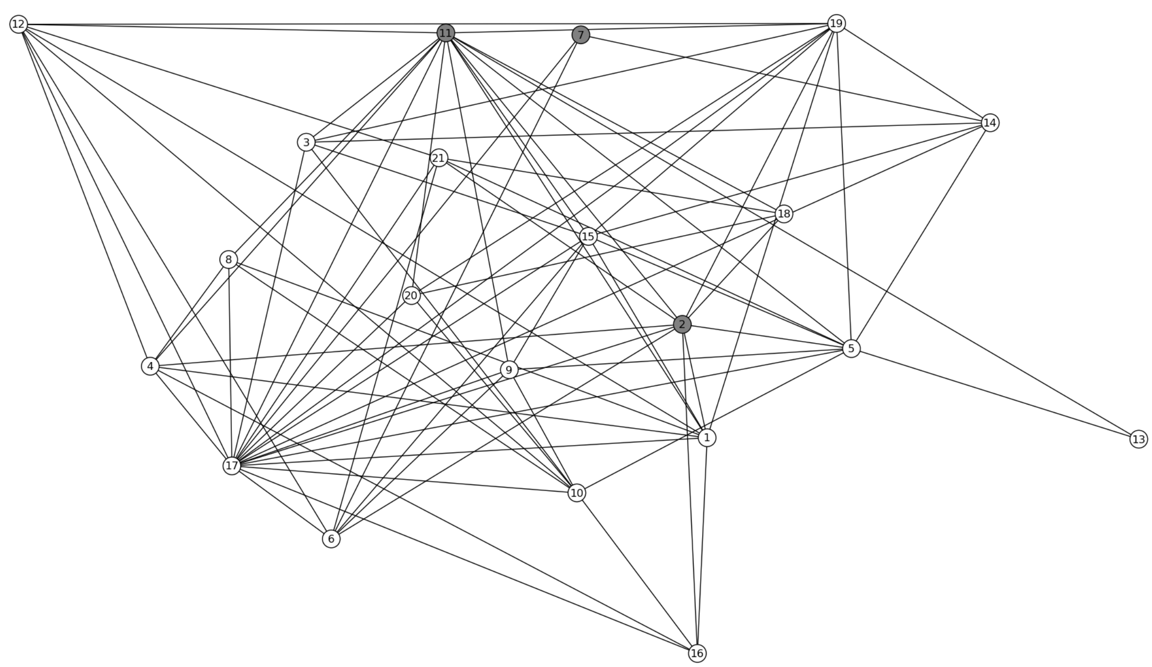

Figure 24: Krackhardt High-tech Managers Friendship network with final key player set highlighted in grey

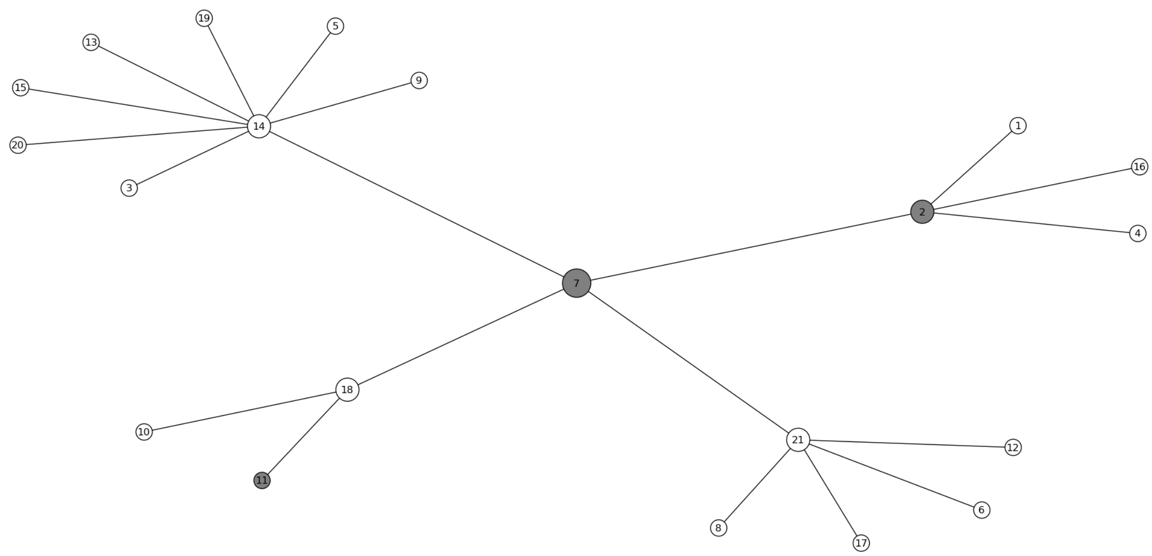

Figure 25: Krackhardt High-tech Managers Job structure network with final key player set highlighted in grey

Table 17: WKPP-Pos scores for key player sets by relationship context

Key Player Set	Advice Seeking	Friendship	Job Structure	Average
{6, 7, 11}	0.10169	0.90000	0.39130	0.46433
{2, 7, 11}	0.10056	0.94737	0.42857	0.49217
{2, 6, 11}	0.09474	0.90000	0.29032	0.42835
{2, 6, 7}	0.10112	0.78261	0.42857	0.43743

4.3 Technique Performance Comparison

In this section the performance of the p-median and hierarchical clustering heuristic are discussed. This is not intended to be a rigorous evaluation of the heuristic, but a discussion of how it performed on the social networks that were analyzed in this chapter.

The p-median technique finds the optimal KPP-Pos and WKPP-Pos key player set for a given social network. The hierarchical clustering technique used in this study never found the optimal solution for the examples in this chapter. However, the results of the hierarchical clustering technique were often near the optimal key player set. The percentage of the optimal solution and timing for each test case is

shown in Table 18. All timing tests were performed on an Intel® i7 920 CPU at 2.67GHz with 6GB of RAM running Ubuntu Linux 10.10 64 bit. The hierarchical clustering technique found a solution that was at least 80% of the optimal in all test cases, achieving a maximum of 97.6% of the optimal solution in one case. Based on the timing results, hierarchical clustering appears to be suited for social networks with a large number of actors and relationships as seen in the Hartford Drug User dataset and Krebs' 9/11 Full Hijacker dataset. It was slower than the p-median technique for the two smaller test cases. For large social networks, much larger than those presented here for these test cases, it is conjectured that hierarchical clustering will be much faster than solving the p-median integer program.

Table 18: Comparison of run times and percentage of optimal for p-median and hierarchical clustering

Case Study	p-median	Hierarchical Clustering	
	Time (seconds)	Time (seconds)	% Optimal
Method's Camp	0.02472	0.03172	95.454
Hartford Drug Users	13.96	3.49	86.727
9/11 Trusted Prior Contacts	0.02640	0.08988	82.758
9/11 Full Hijackers	0.28827	0.14670	97.647

This chapter demonstrated the use of the WKPP-Pos measure in selecting key player sets for a number of social networks from open literature. The impacts of actor and relationship weights to the selection of key player sets was discussed. A technique for applying the WKPP-Pos to a multiple layered social network was developed and demonstrated on a real world social network. In the following chapter, a summary of the development of the WKPP-Pos measure is given. Following that, conclusions about the use of the WKPP-Pos for selecting optimal key player sets is discussed. Finally, suggested future research relating to the advancement of the WKPP-Pos measure is given.

V. Conclusions

5.1 Introduction

This chapter reviews the development of the weighted key player problem positive measure and its application to selected optimal key player sets in social networks. Section 5.2 provides a summary of the development of the WKPP-Pos. Section 5.4 discusses the conclusions about the key player problem and the use of weighted actors and relationships. Section 5.3 outlines potential future research that can be done to expand the WKPP-Pos.

5.2 Summary

The WKPP-Pos measure developed in Chapter III was an extension of the KPP-Pos that was defined by Borgatti. As defined by Borgatti, the KPP-Pos required that the actors and relationships in the social network being analyzed have a weight of unity. In addition, his heuristic approach to finding key player sets did not guarantee an optimal solution. As techniques have been and are being developed to weight social networks, the need for this measure to be extended to include weighted actors and relationships was clear.

The development of the WKPP-Pos required changes to the basic structure of the KPP-Pos formulation, as defined by Borgatti. These changes included restricting the potential weights for actors and relationships to the range $[0, \infty)$. Further, these weights must be real, rational numbers that are additive and multiplicative. The WKPP-Pos measure, ${}^{WD}R$, is defined as the summation of shortest actor weighted distances from the key player set to all other actors in the network. To achieve a normalized measure, ${}^{WD}R'$, the value $n - k$ is divided by the WKPP-Pos measure. This normalized measure can be compared across different social networks to gage the effectiveness of the key player set that has been selected.

Two techniques to find solutions to the WKPP-Pos were developed in this study. The first is the use of the p-median to find optimal solutions to the problem.

The p-median formulation used in this study minimizes the summation of the shortest actor weighted distances from the key player set to all other actors in the network for a given key player set size p. The second technique developed uses hierarchical clustering to form p clusters of actors based on the relationship weights between the actors. The 1-median of each cluster is calculated and the set of p 1-medians is reported as the key player set. The hierarchical clustering technique is a heuristic for the WKPP-Pos. Preliminary results suggest it performs faster than the p-median technique for very large networks of actors.

5.3 Future Research

The following are areas of potential future research related to the WKPP-Pos measure that was developed in this study. The areas range from sensitivity analysis to reformulation of the p-median used to solve for an optimal key player set in a disconnected social network.

A future area of research could be in using sensitivity analysis on actor and relationship weights to determine the range of values that the optimal key player set stays optimal. In the formulation of the p-median problem used in this study, changes to the actor and relationship weights are confined to the objective function. Since the formulation of this problem is an integer program, the duality gap may make the sensitivity analysis difficult. However, understanding the range of weights that the current basis is optimal for can increase the confidence in the optimal solution if there is some doubt in the actual weights for the actors and relationships.

Each actor in a social network has a price; the price for either getting information from them directly or indirectly or the price to influence the actor. Adding the cost associated with each actor for their selection into the optimal key player set would allow for a constraint on the available budget for an operation. The key player set with and without this constraint would allow for a penalty to be calculated based on the available budget. The mixed integer programming formulation of the simple

plant location problem would be a starting point for adding a price for selecting a actor to be in the key player set.

Another extension of the formulation of the p-median used in this study is due to the fact that the p-median formulation in this study is suited for social networks with only one component. The formulation could be extended to allow for finding the WKPP-Pos for social networks with more than one component. Hamill (2006: p. 175–179) discusses a formulation of the p-median that might work for this extension. Further, the formulation of the p-median in this study should work for directed social networks that are strongly connected; however, that concept was not tested. The formulation can be extended to include weakly connected directed social networks.

The implementation of hierarchical clustering in this thesis does not support asymmetric distance matrices. Asymmetric social networks, like those generated by Clark (2005), cannot be analyzed by the clustering technique implemented in this thesis. A different implementation of hierarchical clustering may be able to handle asymmetric social networks. This would be advantageous for large asymmetric social networks as a heuristic to finding a WKPP-Pos key player set. In addition, as heuristics exist that have been developed for the p-median problem, their use should be investigated for larger social networks. They would need to be extended to consider excluded actors.

5.4 Conclusions

The research developed in this study uses actor characteristics, relationship strengths and location theory to identify key individuals in a social network that are strategically located to influence, intercept, strengthen or disrupt data flow between a set of actors. A technique to find the optimal set of actors for a given social network was developed and demonstrated on a number of real world social networks. This extends the tool set of social network analysis to targeting of actors based on actor characteristics, relationship strength and network structure.

From the case studies presented in Chapter IV, it is clear that incorporating actor characteristics and relationship strengths can increase the potential effectiveness of a key player set. The addition of actor and relationship weights allows the analysis to incorporate factors outside of the structure of the social network when determining a key player set or decided between multiple key player sets. In the example in Section 4.2.3, there existed multiple optimal key player sets for the KPP-Pos. When actor weights were incorporated into the analysis, it was found that one of those solutions performed better than the other.

Actor and relationship weights provide additional information about a social network that can and should be leveraged when possible. Although actors in social networks are dependent on one another and their actions are limited by the structure of the social network, each actor has characteristics, independent from the social network, that also limit their actions. Further, not all relationships are equal in strength and that difference should be leveraged when performing social network analysis.

Appendix A. Python Source Code

The code shown in this appendex is the Python code developed and used for all the calculations in this study. Pyhton is a free and open source, cross platform, dynamic programming language that can be found at http://www.python.org. This code makes use of a number of free and open source Python packages: SciPy, NumPy and NetworkX.

Listing A.1: *p*-median, Hierarchical Clustering and WKPP-Pos Code

```
#######################
# p-median generation #
#######################
def pmedian(G,p,path, ineligible=[],h = 0):
  '''pmedian solves the p-median for the inputed network, G.
  It uses lp_solve to solve the mixed-integer problem and
  then parses the outfile. It returns the p medians in a list
  and the clusters as a dictionary.

  Inputs:
  -------
  G:          A NetworkX graph, edge weights allowed with values
              in [1,inf). Edge weights should be stored in
              attribute 'weight' on edges.
  p:          Integer, the number of median vertices to find
  path:       String, path to save the outfile to
  ineligible: (optional), list of nodes that are  ineligible for
              KPP-Pos set
  h:          (optional), a dictionary of node weights, values
              [1.0,inf)

  Outputs:
  --------
  medians:    List, a list of the p medians
  clusters:   Dictionary, a dictionary of the clusters of all the
              nodes in the network. Keyed by the medians.
```

```
'''

import networkx as nx
from subprocess import call
from collections import defaultdict

# Test the inputs and get some basic values #
#############################################
try:
  n = G.number_of_nodes()
except:
  print('ERROR: Input "G" is not a valid NetworkX graph.\...
     nPlease check input format.')
if n == 0:
  print('ERROR: Network provided in input "G" is empty.\nPlease...
      check input.')
  return
sub = 0
if nx.is_connected(G):
  H = G.copy()
else:
  print('WARNING: Network provided in input "G" is not ...
     connected. Using largest component.')
  H = nx.connected_component_subgraphs(G)[0]
  H = nx.convert_node_labels_to_integers(H,discard_old_labels=...
     False)
  #make a conversion table
  convert = H.node_labels.keys()
  old_to_new = {}
  for new in G.node_labels:
    old_to_new[G.node_labels[new]] = new
  sub = 1
n = H.number_of_nodes()
if n == 1:
```

```
    return G.nodes(),{G.nodes()[0]:G.nodes()}
  #make an dictionary of 1.0 if h wasn't supplied
  if h == 0:
    h = dict()
    for node in H:
      h[node] = 1.0
  else:
    #convert the keys of h to their subgraph values
    if sub == 1:
      for node in h:
        h[node] = old_to_new[node]
  if sub == 1:
    ineligible_old =  ineligible[:]
    ineligible = []
    for node in  ineligible_old:
      ineligible.append(old_to_new[ ineligible[node]])
  #distances between nodes
  d = nx.shortest_path_length(H,weighted=True)
  #check that p is an integer, if not convert to integer is ...
     possible or raise error
  if not type(p) == type(1):
    if type(p) == type(1.0):
      print('WARNING: Input "p" is a float, converting to an ...
          integer. This will truncate the value.')
      p = int(p)
    else:
      print('ERROR: Input "p" is not an number.\nPlease check ...
          input.')
      return
  try:
    f = open(path, 'w')
  except:
    print('ERROR: Path provided is not a valid path. Please check...
        input.')
```

```
# write lp file #
################
#objective function
line = 'min: '
for i in d:
  for j in d[i]:
    if not h[i]*d[i][j] == 0:
      value = h[i]*d[i][j]
      line = line+'{0}*x{1}_{2} + '.format(value,i,j)
line = line[0:len(line)-3]+';\n'
f.write(line)
#a demand point can only be served by one supply
for i in d:
  line = ''
  for j in d[i]:
    line = line+'x{0}_{1} + '.format(i,j)
  line = line[0:len(line)-2]+'= 1;\n'
  f.write(line)
#only assign demand points to open supply points
line = ''
for i in d:
  for j in d[i]:
    line = 'x{0}_{1} <= y{2};\n'.format(i,j,j)
    f.write(line)
#only allow p supply points open
line = ''
for i in d:
  line = line + 'y{0} + '.format(i)
line = line[0:len(line)-2]+'= {0};\n'.format(p)
f.write(line)
#eliminate ineligible actors from solution
if ineligible != []:
  line = ''
  for i in ineligible:
```

```
    line = line + 'y{0} + '.format(i)
  line = line[0:len(line)-2]+'<= 0;\n'
  f.write(line)
#variables are binary
#x_ij
line = 'bin '
for i in d:
  for j in d[i]:
    line = line+'x{0}_{1} '.format(i,j)
line = line[0:len(line)-1]+';\n'
f.write(line)
#y_i
line = 'bin '
for i in d:
  line = line+'y{0} '.format(i)
line = line[0:len(line)-1]+';\n'
f.write(line)
f.close()
# Solve the IP using lp_solve #
###############################
command = 'lp_solve {0}'.format(path,path)
output = path+'.out'
f = open(output,'w')
call(command,stdout=f,shell=True)
f.close()
# Read output and report out clusters and medians #
###################################################
f = open(output,'r')
# Check if problem was infeasible
if f.readline() == 'This problem is infeasible\n':
  print('The problem was infeasible. Please check inputs.')
  return
#read in remaining lines. Data starts on line 3 (counting from ...
   0)
```

```
lines = f.readlines()
f.close()
#drop first three entries
lines = lines[3:]
#count keeps track of current line number in lines
count = 0
clusters = defaultdict(list)
for i in d:
  for j in d[i]:
    data = lines[count].split()
    count = count + 1
    if data[1] == '1':
      clus = data[0].split('_')[1]
      node = data[0].split('_')[0][1:]
      clusters[int(clus)].append(int(node))
medians = clusters.keys()
#relabel nodes if orginal network was more than 1 component
if sub == 1:
  #make a copy of old medians
  old_medians = medians[:]
  #reset medians to empty list
  medians = []
  for m in old_medians:
    medians.append(convert[int(m)])
  #Do the same for the clusters
  old_clusters = clusters.copy()
  clusters = dict()
  for c in old_clusters:
    clusters[c] = []
    for n in old_clusters[c]:
      clusters[c].append(convert[n])
  #rekey clusters to use new medians for key
  old_clusters = clusters.copy()
  clusters = dict()
```

```
    for c in old_clusters:
      for m in medians:
        if m in old_clusters[c]:
          clusters[m] = old_clusters[c]
  medians.sort()
  if len(medians) != p:
    print('WARNING: The network could not be partitioned into {0}...
        partitions. Instead, {1} partitions were formed.'.format(...
       p,len(medians)))
  return medians, clusters

########################
# p-cluster generation #
########################
def pcluster(G,p,method='average',h=0, ineligible=[]):
  '''pcluster uses the hierarchy tools in SciPy to create p
  clusters from the input network, G, and then finds the
  1-median of each cluster.

  Inputs:
  -------
  G:          A NetworkX graph, edge weights allowed with values
              in [1,inf). Edge weights should be stored in
              attribute 'weight' on edges.
  p:          Integer, the number of clusters/centers to find
  method:     (optional), a string, the method used to calculate
              the distance between clusters. Valid options: ...
                 single,
              complete, average, weighted, default is average.
  h:          (optional), a dictionary of node weights, values
              [1,inf).
  ineligible: (optional), list of nodes that are  ineligible for
              KPP-Pos set
```

```
Outputs:
--------
centers:   The median for each partition, a list
partition: The p paritions of the network, a dictionary
'''

import networkx as nx
import numpy
from scipy.cluster import hierarchy
from scipy.spatial import distance
from collections import defaultdict

# Test the inputs and get some basic values #
#############################################
try:
  n = G.number_of_nodes()
except:
  print('ERROR: Input "G" is not a valid NetworkX graph.\...
     nPlease check input format.')
if n == 0:
  print('ERROR: Network provided in input "G" is empty.\nPlease...
      check input.')
  return
sub = 0
if nx.is_connected(G):
  H = G.copy()
  H = nx.convert_node_labels_to_integers(H,discard_old_labels=...
     False)
  sub = 1
else:
  print('WARNING: Network provided in input "G" is not ...
     connected. Using largest component.')
  H = nx.connected_component_subgraphs(G)[0]
```

```
  H = nx.convert_node_labels_to_integers(H,discard_old_labels=...
     False)
  sub = 1
  n = H.number_of_nodes()
  if n == 1:
    print('ERROR: Largest component of network provided in ...
       input "G" has only one node.\nPlease check input.')
    return
#make a conversion table
convert = H.node_labels.copy()
new_to_old = {}
for new in H.node_labels:
  new_to_old[H.node_labels[new]] = new
#make an dictionary of 1.0 if h wasn't supplied
if h == 0:
  h = dict()
  for node in H:
    h[node] = 1.0
else:
  #convert the keys of h to their subgraph values
  if sub == 1:
    h_old = h.copy()
    h = {}
    for node in H:
      h[node] = h_old[new_to_old[node]]
if sub == 1:
  ineligible_old =  ineligible[:]
  ineligible = []
  for node in  ineligible_old:
    ineligible.append(convert[node])
#check that p is an integer, if not convert to integer if ...
   possible or raise error
if not type(p) == type(1):
  if type(p) == type(1.0):
```

```
    print('WARNING: Input "p" is a float, converting to an ...
        integer. This will truncate the value.')
    p = int(p)
  else:
    print('ERROR: Input "p" is not an number.\nPlease check ...
        input.')
    return
path_length = nx.shortest_path_length(H,weighted=True)
distances = numpy.zeros((len(H),len(H)))
for u,l in path_length.items():
  for v,d in l.items():
    distances[u][v] = d
# Create distance matrix in proper form
Y = distance.squareform(distances)
# Create hierarchical cluster using method defined in input ...
    method
Z = hierarchy.linkage(Y,method=method)
membership = list(hierarchy.fcluster(Z,t=p,criterion='maxclust'...
    ))
# Create collection of lists
partition = defaultdict(list)
for n,m in zip(list(range(len(H))),membership):
  partition[m].append(n)
if len(partition) != p:
  print('WARNING: The network could not be partitioned into {0}...
      partitions. Instead, {1} partitions were formed.'.format(...
      p,len(partition)))
#Need to find the centers for each cluster now
#C is a dictionary of subgraphs, 1 entry for each cluster
C = dict()
for c in partition:
  C[c] = nx.subgraph(H,partition[c])
# Call pmedian for for each cluster with p = 1
centers = []
```

```
for c in C:
  temp,garbage = pmedian(C[c],1,'pcluster.lp',h=h, ineligible=...
     ineligible)
  centers.append(temp[0])
#rekey partition to use centers for key
old_partition = partition.copy()
partition = dict()
for p in old_partition:
  for c in centers:
    if c in old_partition[p]:
      partition[c] = old_partition[p]
#relabel nodes if orginal network was more than 1 component
if sub == 1:
  #make a copy of old centers
  old_centers = centers[:]
  #reset centers to empty list
  centers = []
  for c in old_centers:
    centers.append(new_to_old[int(c)])
  #Do the same for the partitions
  old_partition = partition.copy()
  partition = dict()
  for c in old_partition:
    partition[c] = []
    for n in old_partition[c]:
      partition[c].append(new_to_old[n])
  #rekey partitions to use new centers for key
  old_partition = partition.copy()
  partition = dict()
  for c in old_partition:
    for m in centers:
      if m in old_partition[c]:
        partition[m] = old_partition[c]
centers.sort()
```

```
  return centers, partition

#########
# kpp_p #
#########
def kpp_p(G,kp,partitions,norm=True,h=0):
  '''Borgatti's KPP-Pos calculation

  Inputs:
  -------
  G:          A NetworkX graph, edge and node weights allowed
              with values in [1,inf). Weights should be stored
              in attribute 'weight' on edges.
  kp:         List, a list of the key players
  partitions: Dictionary, a dictionary of partitions, keyed on kp
  norm:       (optional) True/False, ignores edge/node weights
              and normalizes the KPP-Pos measure
  h:          (optional) Dictionary of node weights, [1,inf)

  Outputs:
  --------
  R:          float, the Borgatti's KPP-Pos measure for the given...
      kp set
              This value is normalized if norm = True
  '''

  import networkx as nx
  from collections import defaultdict

  try:
    n = G.number_of_nodes()
  except:
    print('ERROR: Input "G" is not a valid NetworkX graph.\...
        nPlease check input format.')
```

```
if n == 0:
  print('ERROR: Network provided in input "G" is empty.\nPlease...
      check input.')
  return
sub = 0
if nx.is_connected(G):
  H = G.copy()
else:
  print('WARNING: Network provided in input "G" is not ...
     connected. Using largest component.')
  H = nx.connected_component_subgraphs(G)[0]
  H = nx.convert_node_labels_to_integers(H,discard_old_labels=...
     False)
  convert = H.node_labels.copy()
  sub = 1
  n = H.number_of_nodes()
  if n == 1:
    print('ERROR: Largest component of network provided in ...
       input "G" has only one node.\nPlease check input.')
    return
if type(kp) != type([1]):
  print('ERROR: Input for Key Players is not a list.\nPlease ...
     check input.')
  return
#make an dictionary of 1.0 if h wasn't supplied
if h == 0:
  h = dict()
  for node in H:
    h[node] = 1.0
else:
  #convert the keys of h to their subgraph values
  if sub ==1:
    for node in h:
      h[node] = convert[node]
```

```
#store orignal kp and partitions
orig_kp = kp[:]
orig_partitions = partitions.copy()
#use convert to convert node names
if sub == 1:
  old_kp = kp[:]
  kp = []
  for node in old_kp:
    kp.append(convert[node])
  old_partitions = partitions.copy()
  partitions = dict()
  for p in old_partitions:
    partitions[convert[p]] = []
    for node in old_partitions[p]:
      partitions[convert[p]].append(convert[node])
#Build a dictionary that holds a subgraph of each partition
J = dict()
for p in partitions:
  J[p] = nx.subgraph(H,partitions[p])
inv_dist_sum = 0.0
dist_sub = dict()
for p in J:
  dist_sub[p] = nx.shortest_path_length(J[p],source=p,weighted=...
     True)
for p in dist_sub:
  for node in dist_sub[p]:
    if dist_sub[p][node] != 0:
      inv_dist_sum = inv_dist_sum + (h[node] * float(dist_sub[p...
         ][node]))
if norm:
  R = (float(n) - float(len(kp))) / inv_dist_sum
else:
  R = 1.0 / inv_dist_sum
return R
```

Appendix B. Blue Dart

The following is an op-ed peice, known as a Blue Dart, about the research conducted in this study.

Finding the Right People in a Crowd

— BLUE DART —

Ryan M. McGuire, Capt, USAF*

21 March 2011

Selecting the right people to spread your message in a group is not only about who the selected people know, but also who the selected people are. Marketing companies leveraged this concept when they hire celebrities to endorse a product or campaign. In the field of social network analysis, SNA, the focus has mainly been on studying who knows who in a social network. Capt McGuire wanted to change this by include information about who the people are in a social network and how strong the relationships are between those people.

The social networks being studied are not Facebook, Twitter, or MySpace. Rather, a social network is a group of people, called actors, and their relationships with each other are the focus. Social networking sites like Facebook, Twitter or MySpace are tools that allow you to maintain your personal social network. SNA has tools that allow researchers to analyze social networks and determine which actors are most important to that network.

In his research, Capt McGuire, developed a measure that allows a selection of actors in a group to be rated based on who they are, who they know and how strong their relationships are with the other people. He then developed a technique to identify sets of actors in a group that would produce the highest score for his measure. These actors are the optimal set of individuals in a group to spread a message throughout the rest of the group.

It is often the case that some people in a group may not be willing to help spread a particular message. For that reason, Capt McGuire included a method for excluding certain people from the final set of actors selected by his technique.

Most groups of people change over time. New people come into the group and other people leave the group. Relationships are also formed and broken in groups. The measure developed by Capt McGuire can be used to track the effectiveness of the

*Masters Student, Department of Operational Sciences, Air Force Institute of Technology, Dayton, OH

selected actors as the group they are in changes. If the effectiveness gets too low, a new group of actors may need to be selected to ensure the desired message continues to spread throughout the group.

Capt McGuire's research on this dual use approach has the potential to increase the effectiveness of viral marketing campaigns for companies, or any other strategic or tactical communication effort. When spreading a message about a new product, a marketing company can use Capt McGuire's technique to select a small set of influential people that will reach a large audience. The military can use the technique to select a group of people that will help spread information about humanitarian relief or other operations.

Capt McGuire's research was conducted as part of his graduate degree in Operations Research while he was a student at the Air Force Institute of Technology. He is currently assigned to Air Force Space Command, Peterson AFB, Colorado.

Appendix C. Storyboard

Figure 26 is a storyboard for the research conducted in this study.

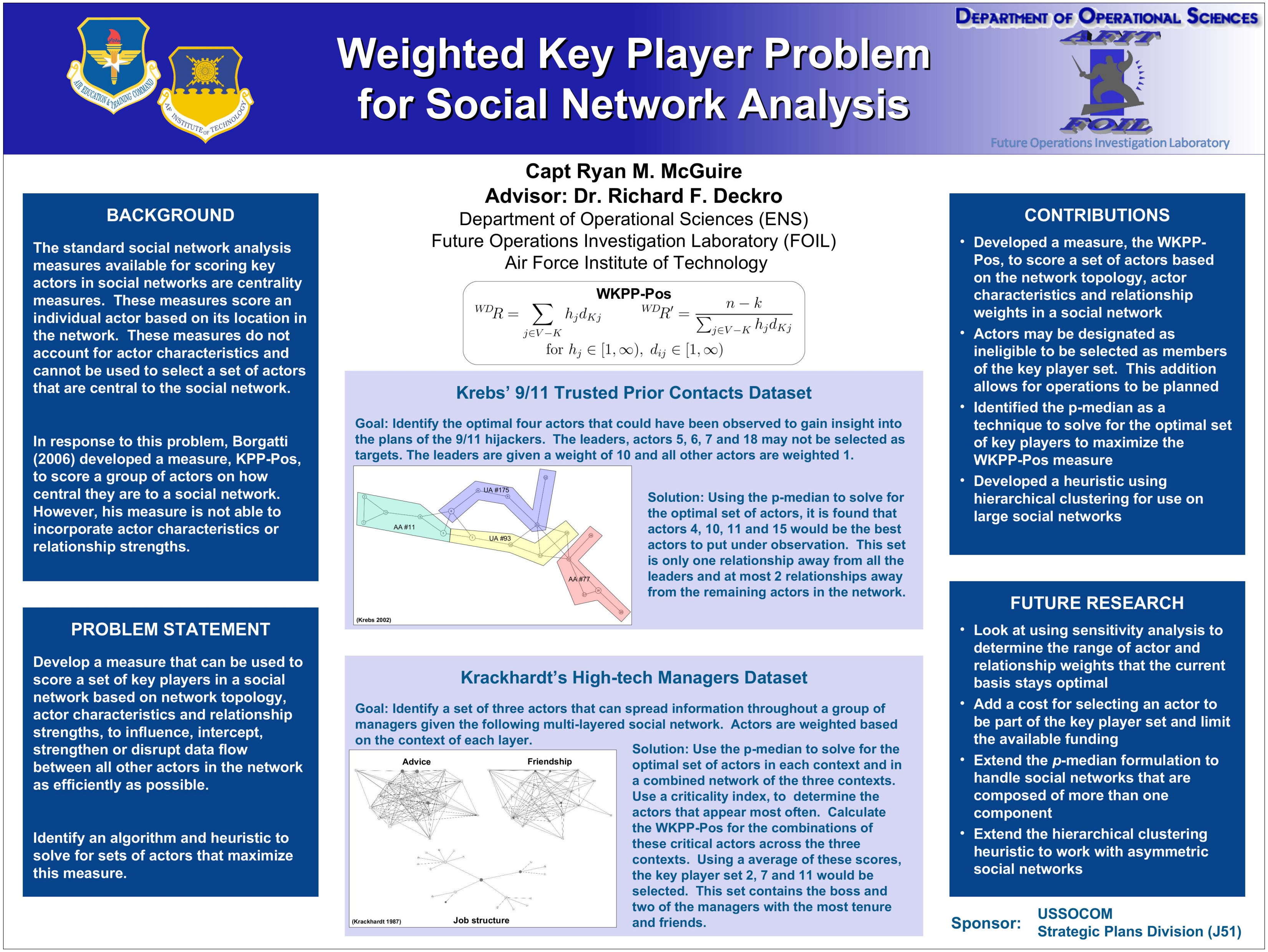

Figure 26: Storyboard

Bibliography

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

Key Player Software. Analytic Technologies, Boston, 2003.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, pages 509–512, 1999.

P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2:113–120, 1972.

P. Bonacich, A.C. Holdren, and M. Johnson. Hyper-edges and multidimensional centrality. *Social Networks*, 23, 2004.

S.P. Borgatti. Centrality and network flow. *Social Networks*, 27:55–71, 2005.

S.P. Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006.

K.M. Carley and D. Krackhardt. A Typology for C2 Measures. In *Proceedings of the 1999 International Symposium on Command and Control Research and Technology*, Newport, RI, June 1999.

K.M. Carley, Dombroski M., M. Tsvetovat, J. Reminga, and N. Kamneva. Destabilizing dynamic covert networks. In *Proceedings of the 8th International Command and Control Research and Technology Symposium*, National Defense War College, Washington DC, 2003.

CASOS. Standard network analysis report: Camp92, 2008. URL `http://www.casos.cs.cmu.edu/computational_tools/datasets/external/camp92/camp92.html`.

C.R. Clark. *Modeling and Analysis of Clandestine Networks*. Master's thesis, AFIT, 2005. Force a change.

R. Collins. *Theoretical Sociology*. Marcourt Brace Jovanovich, 1988.

L.C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

L.C. Freeman. Centrality in networks: Conceptual clarification. *Social Networks*, 1: 215–239, 1979.

J.L. Geffre, R.F. Deckro, and S.A. Knighton. Determining Critical Members of Layered Operational Terrorist Networks. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 6(2):97, 2009.

A.A. Hagberg, D.A. Schult, and P.J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, Aug 2008.

J.T. Hamill. *Analysis of Layered Social Networks*. PhD thesis, AFIT, 2006.

Eric Jones, Travis Oliphant, Pearu Peterson, *et al.* SciPy: Open source scientific tools for Python, 2001. http://www.scipy.org/.

L. Katz. A New Status Index Derived from Sociometric Analysis. *Psychometrika*, 18(1), March 1953.

D. Knoke and R.S. Burt. Prominence. In R. S. Burt and M. J. Minor, editors, *Applied Network Analysis*, pages 195–222. Sage, Newbury Park, CA, 1983.

D. Krackhardt. Cognitive social structures. *Social Networks*, 9(2):109–134, 1987.

V.E. Krebs. Uncloaking terrorist networks. *First Monday*, 7(4), 2002.

V. Marianov and D. Serra. Median Problems in Networks. *Economics Working Papers*, 2009.

E. Minieka. The centers and medians of a graph. *Operations Research*, 25(4):641–650, 1977.

P.R. Monge and N.S. Contractor. *Theories of Communication Networks*. Oxford University Press, Oxford, 2003.

J.L. Moreno. *Who Shall Survive? A New Approach to the Problem of Human Interrelations*. Beacon House, Inc., 1953.

J. Raab and H.B. Milward. Dark networks as problems. *Journal of Public Administration Research and Theory*, 13(4):413–439, 2003.

J.H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

M.R. Weeks, S. Clair, S.P. Borgatti, K. Radda, and J.J. Schensul. Social networks of drug users in high risk sites: Finding the connections. *AIDS and Behavior*, 6 (2):193–206, 2002.

D.B. West. *Introduction to Graph Theory*. Prentice Hall, 2001.

Vita

Capt Ryan M. McGuire graduated from the Eden Prairie High School in Eden Prairie, Minnesota. He entered undergraduate studies at the United States Air Force Academy in Colorado Springs, Colorado where he graduated with a Bachelor of Science degree in Physics in May 2002. He was commissioned through the United States Air Force Academy.

His first assignment was at Kirtland AFB where he served as a polarization research simulation physicist in August 2002. In June 2005, he was assigned to the Space Superiority Systems Wing, Los Angeles AFB, California, where he served as a space control technology project officer and a space situational awareness program manager. In January 2008 he was appointed an acquisition branch chief in the Space Superiority Systems Wing. In August 2009, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to Peterson AFB, Colorado.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
03-24-2011	**Graduate Research Thesis**	Aug 2009 – Mar 2011

4. TITLE AND SUBTITLE	
WEIGHTED KEY PLAYER PROBLEM FOR SOCIAL NETWORK ANALYSIS	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) McGuire, Ryan M., Captain, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765	AFIT-OR-MS-ENS-11-13
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USSOCOM Strategic Plans Division (J51) August G. Jannarone 7701 Tampa Point Blvd. Bldg. 501A, Rm. 121, J51 MacDill AFB, Florida 33621	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Social network analysis is a tool set whose uses range from measuring the impact of marketing campaigns to disrupting clandestine terrorist organizations. Social network analysis tools are primarily focused on the structure of relationships between actors in the network. However, characteristics of the actors, such as importance or status, are generally the output of the social network analysis rather than an input. Characteristics of actors can come from a number of sources to include information gathering, subject matter experts or social network analysis. Further, the strength of relationships between actors in social networks are often assumed to be all equal. However, relationships range from strong familial like relationships to weak casual relationships. The research developed in this thesis uses actor characteristics, relationship strength and location theory to identify key individuals in a social network that are strategically located to influence, intercept, strengthen or disrupt data flow between a set of actors. In this technique, actor characteristics and relationship strength are used as inputs into the analysis and the output is a set of actors which satisfies the desired objective and the constraints of the given problem. This extends the tool set of social network analysis to targeting of actors based on actor characteristics, relationship strength and network structure.

15. SUBJECT TERMS

Social network analysis, key player, actor weights, relationship weights

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Richard F. Deckro, (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4325; e-mail: richard.deckro@afit.edu
U	U	U	U	107	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18